MICROBYTE

Vol II N° 8

TODO COMPUTACION

DICIEMBRE 1985
N.° 19 $ 180



Proyecto Camino

Ejecución de Programas MS-DOS

Programación de Trabajos y Máquinas

Programas Atari, Commodore, Sinclair

Hay muchos computadores personales HP 150 un *"muy personal"* computador.

HEWLETT PACKARD

Microcomputador (PC), con Procesador Intel 8088-2 de 18 Bits y 8 MHz Sistema Operativo MS-DOS 2 11 – Memoria de 256 Kb expandible a 640 Kb – Disco flexible doble de 31/2" con 2 unidades de 710 Kb c/u formateados. Disco Winchester de 15 Mb y Disco flexible de 31/2, con 710 Kb formateados. Toque Mágico exclusivo, para seleccionar las aplicaciones directamente. Software standard: Lotus 1-2-3, D-base, Archivo Electrónico, Memomaker. Software con aplicaciones netamente nacionales y de acuerdo a sus requerimientos



OLYMPIA (Chile) LTDA Av. Rodrigo de Araya 1045 – Macul



ARICA	IQUIQUE	ANTOFAGASTA	V. DEL MAR	VALPARAISO	SANTIAGO	RANCAGUA	TALCA	CONCEPCION	TEMUCO	OSORNO	PUNTA ARENAS
61 5 74	2 66 64 2 31 94	22 23 26	66 07 30	21 67 72	22 55 044 66 22 63	2 63 15	3 66 11	2 17 02	3 17 03	66 66	2 15 37 2 51 34

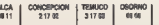




Foto Portada

El encuentro con el futuro ya está aquí ¿Lo sabremos aprovechar ?

Director Responsable
Jorge Carrera R

Coordinador General
Jose Kalfman T

Director Publicidad y RR PP
Ariel Leporatti P

Ventas
Orlando Zepeda

Directora de Arte
Paz Barba

Montaje
Rodolfo Hilmer

Fotografia
Gentileza USIS

Cuerpo Editorial
Jaime Aravena
Jorge Cea
Carlos Contreras

Corresponsales en el exterior
Luis Kalfman T (Londres)
Alfredo Zarowsky (Paris)
Victor Kahan (Ohio)

Fotocomposición
LASER

Representante Legal
Jorge Carrera R
Direccion Merced 346-Of F
Fono 393866

Distribucion
Antartica S A

Impresion
Impresora Nacional quien sólo actúa como impresor

Microbyte es una publicacion mensual de KVC Asociados

Microbyte no puede asumir ninguna responsabilidad por errores en artículos, programas o avisos publicitarios

Las opiniones expresadas en estas páginas [illegible] sus autores y no representan necesariamente el pensamiento de sus editores.

Colaboraciones de los lectores son bienvenidas y serán publicadas previa revisión con un pago de acuerdo a tipo de colaboración y calidad

Las colaboraciones deben venir [illegible] impresas [illegible] acompañadas de material gráfico. En el caso de listados de programas mayores de 15 líneas es preferible enviar cassette o disco [illegible]

SUBSCRIPCIONES

Valor subscripciones semestral (6 Ejs.)

Correo Certf. Stgo. y Prov	$ 1.050
Entrega por mano Stgo	$ 950

Valor subscripciones anual (12 Ejs.)

Correo Certf. Stgo. y Prov	$ 1.950
Entrega por mano Stgo	$ 1.800

Solicite un representante al fono 393866 en Merced N° 346 Of F Santiago Chile

Vol II N° 8

Muchas veces, desde estas mismas páginas y en diversas otras publicaciones, se ha mencionado la importancia que podría tener el desarrollo tecnológico en el área informática para el desarrollo de los países.

Por esto, sería reiterativo volver al tema mostrando ejemplos de como en otros países, esta idea ha tenido una cabal concretización en la elaboración de proyectos nacionales y en la creación de organismos ministeriales abocados específicamente a esta área.

Sin embargo, hay algunos episodios curiosos que permiten apreciar desde otras perspectivas la importancia del fenómeno informático.

En efecto, en las costas normandas, en Francia, bajo la dirección del general Jean Becam, comandante retirado de la fuerza aérea de su país, se está organizando una nueva resistencia. El enemigo esta vez, no utiliza bombarderos ni misiles, siendo su arma principal y más mortífera las palabras: la jerga computacional en inglés.

De acuerdo al general Becam, utilizar términos como "byte", "bug" o "network" significa rendirse ante el lenguaje del conquistador y para su defensa ha creado la Asociación de Especialistas en Computación Franco-parlantes como herramienta para defender la supervivencia de la cultura gala.

A pesar de sus esfuerzos, mediante los cuales han sido creadas una larga lista de palabras en francés para reemplazar los términos ingleses, los operadores franceses procesan aún "un batch de data" o hacen "un dump" en "le floppy".

Para países como Francia, no se trata solamente de un aspecto cultural o de orgullo nacional. Francia es un país industrializado para el cual cada palabra en inglés que acepta en su vocabulario representa a las cuantiosas mermas económicas que le significa la pérdida de terreno frente a Estados Unidos.

La razón de esto es simple. Si el idioma que se utiliza preferentemente es el inglés, significa que la ciencia computacional también es de procedencia norteamericana y los computadores franceses una copia barata de éstos, así como el champagne de California.

No deja de parecer interesante por tanto, la actitud que ha tenido IBM en Francia, tratando en todo momento de parecer el más auténtico francés obviando así el natural rechazo de los franceses por todo lo que no sea de su país. En efecto, al liberar en Francia los Personal Computer de IBM, los famosos PC como son llamados en todo el mundo, éstos fueron denominados los "O P" por la traducción francesa de computador, "ordinateur".

Lo paradojal, es que a pesar de todos los esfuerzos de IBM y del general Becam, los franceses no han asimilado esa terminología nacional y se refieren libremente a los "PC" al igual que sus colegas del resto del mundo. Como última curiosidad, la propia palabra "ordinateur", tan cara a los puristas de la lengua francesa, fue acuñada por la propia IBM, la que ya en 1958 acudió a un comité de la Sorbonne para que ideara un término francés para los "computers" que comercializaba IBM.

Volviendo a nuestro país, no es fácil definir cuál debiera ser la actitud correcta frente a la invasión cultural, pero de seguro es un factor que debiera ser seriamente analizado. No se trata solamente de una terminología en idiomas extraños que dificulta aún más el dominio de las ciencias de la computación. Hay además aspectos valóricos, éticos, modelos de desarrollo y un sinfín más que entran de contrabando junto a la voluminosa cantidad de computadores que se están instalando en Chile.

Huelga reiterar frente a todo lo anterior el desconcierto que se produce cuando en lugar de crearse instancias de reflexión y diseño de estrategias de desarrollo, van desapareciendo los pocos organismos estatales capaces de hacerlo.

NOTICIAS

Ericsson Portable PC

Elegante como los productos suecos pero fabricado en Japón, el Ericsson Portable PC es probablemente uno de los equipos con mas hermoso diseño

Basado en un procesador 8088 y MS-DOS 2 1, este equipo puede correr la mayor parte de las aplicaciones de un IBM PC Su resolución es de 640 por 400 pixeles y tiene la ventaja sobre otros portátiles de venir con pantalla de gas plasma

Standard viene con 256K de memoria expandibles a 512 una diskettera de 5 25 con capacidad para 360K puertas paralela y serial Opcionalmente se le puede incorporar un disco RAM de 360K, una impresora integrable y modem interno



Inteligencia Artificial en el Macintosh

Neuron Data, una pequeña empresa francesa instalada en Palo Alto California, anuncio un generador de sistemas expertos para el Macintosh durante la ultima convención de la asociación americana por la inteligencia artificial

Programado originalmente en Lisp y luego en Assembler Nexpert corre en un Macintosh con 512K y contiene un motor de inferencia lógica que permite graficar en todo momento la concatenación de reglas construidas por el sistema en forma de malla, lo que permite al investigador visualizar cómo van siendo afectadas las reglas y cuales resultados dependen de ellas con cada modificación

Este programa, junto a otros de Teknowledge y Texas Instruments están demostrando la vertiginosa caida en los precios de estas herramientas de quinta generacion (cinco mil dólares), las que están haciendo su entrada al mundo de los microcomputadores

El turno del modem

En el numero anterior comentabamos respecto al verdadero asalto que está lanzando IBM hacia diversos sectores de productos afines a la computación Esa vez comentábamos respecto a su nueva linea de impresoras y las redes de datos en Japón

Esta vez IBM nos sigue sorprendiendo con la introduccion de su primer modem para microcomputadores El unico modem de esa naturaleza jamás hecho por IBM fue el modem interno con que venia el ya discontinuado PC Junior Hasta ahora IBM distribuia por sus canales de distribucion la linea de modems fabricados por Hayes

Los modems introducidos por IBM (fabricados para este por Rockwell International), son una tarjeta modem Asincrona a 1 2 kbits seg para insertar en el PC por US$ 499 Un modelo independiente también de 1 2 kbits/seg puede funcionar igualmente en modo sincrono y se vende en US$ 609

IBM se querella contra Fujitsu

IBM acusó judicialmente a Fujitsu de copiar ilegalmente el sistema operativo MVS/XA que corre en una nueva linea de mainframes IBM por lo que estaria reclamando una reparacion multimillonaria

Fujitsu que se ha convertido en uno de los principales fabricantes de mainframes, se ha especializado en equipos compatibles con IBM, para lo cual habia llegado a un acuerdo con este para utilizar su software operativo previo pago de royalties

Sin embargo, este acuerdo de 1983 no contemplaba a este nuevo sistema operativo por lo que esta acción de IBM podria convertirse en un grave revés económico para la gigantesca compania japonesa

De prosperar esta demanda, seria la mayor desde 1983, en que en un bullado caso, agentes del FBI haciéndose pasar por empleados de IBM capturaron a altos ejecutivos de Hitachi y Mitsubishi tratando de obtener software operativo de IBM



COMPUTADOR PERSONAL HP 150 II DE HEWLETT–PACKARD

EL COMPUTADOR DE EMPRESA POR DEFINICION.

[illegible] es [illegible] do

ASC le ofrece [illegible] computador [illegible] HP 150 que fue diseñado [illegible] pensando principalmente en dar soluciones [illegible] necesidades de las empresas

EL HP 150 le ofrece

- Gran facilidad de uso [illegible]
- Instrucciones y manuales [illegible]
- [illegible] similares [illegible] para solucionar [illegible] requerimientos de su empresa
- Confiabilidad del equipo
- Avanzada tecnología
- Seguridad de permanencia en el tiempo de la [illegible] respaldo [illegible] liderazgo tecnológico de Hewlett [illegible]
- Y por supuesto probados y [illegible] programas administrativo [illegible] cuentas corrientes remuneraciones [illegible] inventario contabilidad y otros además de [illegible] populares programas de procesamiento de palabras y análisis financiero como [illegible] Lotus 1 2 3 [illegible] etc

[illegible] son muchas y cada día más las empresas [illegible] que optan por la solución HP 150 [illegible] seguridad y respaldo que ASC brinda a sus usuarios de acuerdo a las normas internacionales de [illegible] y soporte de Hewlett Packard

Decídase hoy por la solución ASC y aumente la productividad y competitividad de su empresa

EN COMPUTACION... ASC Y HEWLETT–PACKARD... ES SUPERIOR.

DESDE US$ 4 000 e q m/n + IVA

ASC futuro con experiencia



REPRESENTANTE OFICIAL PARA CHILE DE LA LINEA COMPLETA DE COMPUTADORES HEWLETT PACKARD

AUSTRIA 2041 – PROVIDENCIA SANTIAGO FONOS 2235946 • 2236148 • 744780 – TELEX 340192 ASC–CK

DISTRIBUIDORES AUTORIZADOS TASCO LTDA MAC IVER 105 TELS 2512289 2512288 EN SANTIAGO DISEÑO Y DISTRIBUCION DE SISTEMAS LTDA MERCED 280 OF 42 TEL 393391 EN SANTIAGO SISTEMAS COMPUTACIONALES TAURUS COMPUTERS LTDA CAUPOLICAN 567 OF 94 TEL 25515 EN CONCEPCION FIRMANI Y CIA LTDA MANUEL MONTT 730 TEL 34239 EN TEMUCO Y RAMIREZ 870 TEL 5757 EN OSORNO

DISEÑO & COMUNICACION

SPSS/PC

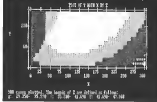
Hasta hace poco tiempo los estadisticos que deseaban procesar sus datos debian recurrir a un mainframe, escribir sus propios programas en Fortran o utilizar el famoso SPSS (Statistical Package for the Social Sciences)

SPSS PC es una versión de ese mismo popular programa diseñado para correr en un IBM PC, XT o AT Requiere de un disco fijo y mínimo 320K de RAM además de una infinita paciencia o experiencia con la versión para mainframes pues es poco amistoso con el usuario (El manual es de nada menos que 624 páginas)

Los programas ocupan casi 3 mega en el disco fijo y para correr debe mantener inserto en la diskettera un disco que actua como la proteccion del paquete

Al utilizarlo SPSS asume que uno tiene algun software para ingresar datos un manejador de base de datos o conexión a un mainframe SPSS acepta archivos ASCII de hasta 1024 caracteres por registro o 200 campos por registro

A un valor de US$ 795 este paquete seguramente se va a convertir en un superventas entre aquellos estadisticos que necesitaban una herramienta similar y no tenían acceso como es lo usual a computadores de mayor envergadura



El proyecto Esprit queda sin fondos

Dieciocho meses después de su lanzamiento el proyecto Esprit en el que diversos paises de la comunidad economica europea se comprometian a financiar en conjunto investigaciones en el area de la informatica tocó fondo el presupuesto (unos 900 millones de dolares) que inicialmente se habia proyectado para cubrir cinco años de investigación

Para sus organizadores este balance, sin embargo no es negativo, pues, segun ellos el desarrollo de este fue mucho mas veloz que lo estimado En efecto hay 173 proyectos en ejecución en los que participan 448 instituciones entre empresas privadas universidades e institutos de investigacion

El propósito de este proyecto es crear en Europa una infraestructura tecnologica capaz de competir en el largo plazo con Estados Unidos y Japón Para Europa es tan importante el terreno de la alta tecnologia que incluso, de escasear los fondos otorgados para Esprit la comunidad estaria dispuesta a retirar fondos del proyecto Eureka, otra iniciativa europea pero a nivel de industrias para volcarlos a Esprit con su tecnologia de punta

Quimica y computación

Los avances en computación grafica han permitido que ultimamente un nuevo rubro utilice con mayor provecho las herramientas informáticas

En efecto, los principales productores de productos quimicos, sean farmaceuticos agricolas o industriales están utilizando las pantallas de sus terminales para diseñar y simular el comportamiento de complejas moléculas

Entre los usos típicos de la computación en esta área está el diseño inicial de productos Con la facilidad de poder alterar el ordenamiento molecular en la pantalla los cientificos son capaces de elegir las estructuras con las mayores posibilidades de lograr el resultado requerido Además a partir de una sustancia dada un científico puede optimizar su estructura modificando la posición de los átomos en la molécula viendo diversas alternativas sin necesidad de realizar costosos experimentos fisicos

Este ultimo uso es utilizado fundamentalmente para violar legalmente patentes de productos, pues al cambiar la posición de algunos átomos claves dentro de una molécula se puede generar una molecula con cualidades similares pero sin incurrir en violacion de patentes



[illegible]

Nueva red SWIFT

Para mediados de 1986 fue anunciada la puesta en marcha de una version mejorada de la red interbancaria SWIFT (Society for Worldwide Interbank Financial Telecomunications)

Creada en 1973 por 239 bancos de 15 paises como una institución cooperativa dedicada a satisfacer las necesidades de transmisión de transacciones interbancarias, Swift ha crecido hasta contar hoy con 1257 bancos asociados en 5 paises

Uno de los logros más importantes de Swift ha sido crear standards y formatos que permiten a los bancos comunicarse claramente y sin ambiguedades

La nueva versión de Swift surge del avance tecnológico en el área de transmisión de datos y de su propio crecimiento como institución Entre las ventajas del nuevo sistema se destaca una mayor flexibilidad por su arquitectura modular descentralizada, una mayor capacidad y velocidad Incluso, esta versión cuenta con la posibilidad de consultas interactivas y su capacidad inicial será de un millón de transacciones dianas

Los terminales a utilizar por el nuevo sistema serán los ST400 de Digital a diferencia de los anteriores que eran Burroughs

Siguen las travesuras

Veintitrés hogares en San Diego, California, se vieron abruptamente expropiados de sus microcomputadores hogarenos luego que agentes del FBI descubrieron a un grupo de hackers que habian hecho de las suyas en la base de datos de los computadores del Chase Manhattan Bank

De acuerdo a voceros del banco, los intrusos sólo lograron burlar el primer nivel de seguridad de sus sistemas lo que no les permitia girar fondos o conocer detalles de los nombres y balances de los cuentacorrentistas

Sin embargo, si fueron capaces de cambiar algunas claves de acceso impidiendo a los usuarios del sistema accesar sus propios servicios

Para identificar a los delincuentes, el FBI rastreó todas las llamadas a la base de datos llegando de ese modo a los autores quienes en castigo se han quedado sin poder disfrutar de PacMan o Space Invaders y obligados a leer algun buen libro

Sperry anuncia AT - Compatible

Sperry Corp uno de los principales fabricantes norteamericanos de mainframes, anunció la proxima aparicion de un modelo de microcomputador AT-Compatible

El equipo fabricado originalmente por Mitsubishi Electronics en Japón es el segundo modelo de computador personal que introduce Sperry Anteriormente habia lanzado un PC Compatible tambien fabricado por Mitsubishi

Lo más interesante en esta información es que junto al nuevo equipo, Sperry liberara un PC Link para integrar al nuevo equipo al Sperrylink un sistema integrado de oficina que incluye procesamiento de texto computacion personal almacenamiento masivo de datos y telecomunicaciones

En efecto cuando gigantes de la computación como Sperry adoptan los computadores personales como parte de sus sistemas de automatizacion de oficinas, no hacen mas que reafirmar el importante papel que han comenzado a jugar los computadores personales en las empresas comparado con sus inicios hace algunos años en que solo eran considerados como instrumentos educacionales y recreativos o como hobby de electronicos en sus garages

Digital Research modifica GEM

Concebido como un sistema operativo capaz de emular la interfaz hombre-máquina del Macintosh en otros equipos, GEM (Graphics Environment Manager) creado por Digital Research fue formalmente acusado por Apple por plagio y deberá ser modificado.

Si bien Digital Research en principio se declaró no culpable al fin debió llegar a un acuerdo amistoso con Apple evitando así las consecuencias de un fallo judicial probablemente adverso. De haberse producido éste, Digital Research debería haber pagado las indemnizaciones correspondientes, retirar todas las copias que se encuentran en circulación, licenciadas a diversos fabricantes de equipos.

En este acuerdo amistoso Digital se comprometió a pagar una suma no revelada a Apple modificar sustancialmente tres módulos (Gem Paint, Gem Draw, Gem Desktop) y retirar de sus campañas publicitarias toda referencia al Macintosh de Apple.

Kaypro 16

Kaypro, anteriormente conocido por sus sistemas CP/M transportables tipo Osborne se acopló a la liga de los PC Compatibles con un modelo transportable que incluye un disco fijo de 10 mega y diskettera de 360K.

Al igual que el IBM PC el Kaypro 16 viene con un procesador 8088 a 4.77Mhz, 256K de RAM expandibles a 640 y acepta un coprocesador 8087 opcional. El monitor incorporado es de 9 pulgadas y su resolución es de 640 por 200 pixeles.

Una de las características que hicieron popular a Kaypro en su versión de 8 bits fue la abundancia de software con que dotaba a sus equipos al venderlos. Esta misma característica se repite en el nuevo modelo aunque no sean los paquetes más cotizados.

En efecto con el Kaypro 16 se entrega WordStar, Mailmerge, CalcStar, InfoStar y un paquete de comunicaciones Mite.



Sistema Texas de Inteligencia Artificial

Texas Instruments liberó recientemente un nuevo computador dirigido principalmente al desarrollo de aplicaciones en inteligencia artificial. Conocido como el Explorer, este computador fue diseñado utilizando licencias del Instituto Tecnológico de Massachussets y de Lisp Machine Corporation.

Las aplicaciones en inteligencia artificial trabajan en base a símbolos: los que pueden ser objetos, relaciones, propiedades o conceptos los que son analizados de acuerdo a determinadas reglas. Para esto, se requiere de poderosos equipos en términos de velocidad y memoria a fin de lograr algún tipo de resultados en un tiempo razonable. El Explorer, está basado en un procesador Lisp, cuenta con dos megabytes de memoria y capacidad para 140 Mb en disco Winchester.

Disco duro tamaño tarjeta

HardCard es el nombre de un nuevo producto desarrollado por Plus Development Corp., una subsidiaria de Quantum Corp. mediante el cual se incorpora un disco duro, con capacidad de 10 mega en un slot de expansión de un IBM PC o compatible.

El tamaño de este periférico, 4 por 13 pulgadas por una pulgada de espesor, es el disco duro más pequeño jamás fabricado, consume sólo 11 watts y pesa menos de un kilo.

Adicional al tamaño, tiene la ventaja de venir con su propio software de inicialización y un administrador de archivos más sencillo de utilizar que el propio MS-DOS.

Computación para no-videntes

De acuerdo a informaciones proporcionadas directamente por el director del laboratorio de Socio-Informática de la Universidad de Cuyo en Mendoza investigadores de esta casa de estudios trasandina, desarrollaron un programa para facilitar el acceso de no-videntes al mundo de la informática.

El programa que corre en microcomputadores de bajo costo utiliza un sintetizador de voz común, mediante el cual cada vez que es presionada una tecla el computador emite una voz que la identifica.

Para adiestrar a los ciegos en el manejo del teclado el programa incluye rutinas de práctica consistentes en juegos en que el usuario debe adivinar números y el computador responde mayor o menor, operaciones matemáticas, etc.

NCR



Estamos solamente en grandes proyectos. Por eso estamos muy cerca de usted.

Cuando usted opera el cajero automático de su banco, está operando un equipo de computacion NCR

¿Le sorprende?

Es que NCR quiere estar presente, muy cerca suyo, simplificándole la vida

Cerca dal 80% de los bancos que poseen Cajeros Automáticos en Chile usan Cajeros NCR

Y este liderazgo absoluto en ATM (Automated Teller Machine) es producto de la innovadora tecnologia computacional de NCR

NCR

Donación de IBM a Hospital del Salvador

Con equipos computacionales que ayudarán a racionalizar y mejorar la productividad en las áreas de farmacia y de atención a pacientes, contará próximamente el Hospital del Salvador. La incorporación de estos elementos se materializará al suscribirse un convenio por el cual la empresa IBM de Chile donará a la institución de salud, equipos, programas y apoyo por un monto superior a los 200 mil dólares

El aporte de IBM consiste en equipos de avanzada tecnología (un sistema /36, con almacenamiento de 600 megabytes, 15 terminales y tres impresoras), el uso de licencias de programas IBM (software), y el desarrollo de algunos programas especiales para su empleo en el Hospital. Se estima que todos los sistemas estarán operativos en los próximos meses

Liberan el DS 500



En el curso del presente mes debe ser liberado por Lógica el DS 500, línea de equipos PC compatibles con el sistema operativo MS DOS 3 1 y con todo el software para ese sistema (planillas Lotus 1-2-3, WordStar etc) Además puede ser usado como terminal de equipos más grandes de la línea MAI Basic Four

Los modelos a liberar son el 500 con dos unidades de diskette de 360 K cada uno, el 510 con 10 MB en disco y el 520 con 20 MB Su precio de venta aun no había sido fijado al cierre de esta edición



El stand de Lógica en la FISA exhibió el microcomputador MAI 2000 y herramientas agregadas a él como el generador de programas de cuarta generación Origin el Magnet para redes de computadores y el Harmony que permite incorporar el IBM PC como terminal de los MAI

En la foto ante su stand aparecen de izquierda a derecha Mario Trattan gerente de ventas de Lógica Alberto Urriksen gerente de Marketing Ernesto Ortega ejecutivo peruano y Juan Enrique Royse gerente general de Lógica



Nuevo local de Informa

En los primeros días de octubre, la Compañía de Informática Nacional Informa Ltda., inauguro un nuevo local de atención directa al público

Informa pone a disposición del usuario insumos computacionales como Diskettes, Cintas Impresoras, Cintas Magnéticas, Carpetas Colgantes y otros

Ubicada en la Galería Crillón 27 (Agustinas 1035) ofrecerá un servicio eficaz a la creciente demanda computacional

Ventas

La Compañía Chilena de Lubricantes adquirió a Elca-Computación un microcomputador Altos 986T 80 compuesto por un MB de memoria RAM, 80 MB en disco, 60 MB en cinta magnética, tres terminales y dos impresoras de 180 CPS El equipo se encuentra instalado y es utilizado en toda la gestión administrativa de la empresa

Lógica colocó un equipo MAI 2000 en la firma Conavicoop En su configuración figuran 44 mega en disco, tres pantallas, 1 MB de memoria e impresora Los compradores lo utilizaran en aplicaciones administrativas

La firma Transmar compró igual equipo con la misma configuración y para similar aplicación

Un microcomputador Casio FP6000-S adquirió la agencia de aduana Leonardo Tamblay Flores Componen el equipo 256 KB de memoria RAM, dos drives 320 KB c/u, un disco de 10 MB, pantalla e impresora de 160 CPS

Insólito compro a Elca un microcomputador Casio FP6000-S compuesto por 256 KB en memoria RAM dos drives de 320 KB cada uno, una pantalla y una impresora de 180 CPS

Seminario CLAT "Nuevas Tecnologías"

Pedro Sanchez A Presidente del Sindicato ECOM participó como relator en un Seminario organizado por la CLAT en Buenos Aires

El tema que expuso fue Nuevas Tecnologías y su impacto en el mundo laboral se analizó in extenso los fenómenos originados por la Computación e Informática, Robótica Control Automatico y Cibernética, su vinculación con la cesantía estructural sector informal de empleo, y su crecimiento y materias afines involucradas con las Relaciones Industriales

NUEVOS PC's

Multitech



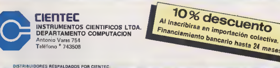


Ahora a su alcance toda una línea de Computadores MPF–PC, compatibles con programas, tarjetas y accesorios IBM–PC.®

MODELOS	MPF–PC POPULAR	MPF–PC	MPF–PC/XT	MPF–PC PLUS
Microprocesador	INTEL 8088 de 16 bits			INTEL 8088–2
Entrada/Salida	1 Puerta Paralela CENTRONICS 1 Puerta Serial RS–232 C			
Velocidad Procesador Coprocesador 8087	4.77 MHz ----	4.77 MHz opcional	4.77 MHz opcional	8 MHz opcional
Memoria ROM	8 KB expandible a 48 KB			
Memoria RAM	256–512 KB	640 KB	640 KB	640 KB
Disketera 360 KB Disco Fijo	1–2 ----	2 ----	1 10–20 MB	1–2 10–20 MB
Conectores disponibles	1	4	3	3
Tarjetas video MDA	Monocromatica texto de alta resolución			
Tarjetas video CGA	Color texto baja resolucion y graficos			
Tarjetas video MGA	Monocromatica texto y graficos de alta resolucion			
Precio desde	* US $ 1.524. + IVA	* US $ 2.480. + IVA	* US $ 3.840. + IVA	4.140. + IVA

(* Equivalente moneda nacional)

BARTOV



Una buena información = una buena decisión

Si su empresa adquirió el mes pasado un total de 100 diskettes para que funcionara correctamente el sistema computacional de almacenamiento de datos, usted probablemente canceló un valor de $ 148 400. Si usted hubiese recurrido a SERVIPRES, firma especializada en estudios de precios, cotizaciones y otras investigaciones de mercado probablemente pudo ahorrar $ 99 400. Una suma interesante que no habría que menospreciar.

Según un estudio de mercado de los diskettes, realizado por esta firma se pudo comprobar a través de los precios de venta al público que tienen los propios oferentes, resultados interesantes. Primero, es conveniente hacerse asesorar antes de comprar, especialmente cuando los resultados como éstos revelan cómo pueden reducirse substancialmente los costos. Y luego, que existe una amplia gama de marcas distintas, las cuales ofrecen los mismos servicios por un artículo similar.

SERVIPRES continuará ofreciendo a través de este espacio un servicio informativo gratuito para los suscriptores de esta revista, con respecto a la amplia variedad de precios que presentan en el mercado nacional, toda la línea de computadores que se ofrecen, como también todos los elementos y accesorios que son indispensables para su funcionamiento.

En el cuadro siguiente se observan los resultados del estudio comparativo realizado recientemente, al mercado de diskettes.

MARCA DISKETTE	CERTRON		VERBATIM				MEMOREX			3M		XIDEX		TDK		NASHUA	BASF	ELEPHANT	IBM	DYSAN	MAXELL
Locales Encuestados	Local 1	Local 2	Local 3	Local 4	Local 5	Local 6	Local 3	Local 8	Local 9	Local 8	Local 11	Local 8	Local 7	Local 2	Local 3	Local 3	Local 2	Local 7	Local 10	Local 11	Local 6
Una cara Doble densidad Soft sector	560	490	612	652	744	543	552	630		644	543	666	980	–	990	–		580	1 484	980	543
Dos caras Doble densidad 5¼ Soft sector	600	580	744	772	652	723	672	780	885			836	700	798	756	648	590	790	1 464	996	–
Dos caras 16 sectores (Hard-sector)			1 140	772	1 284		672	780	–	–		636	–	–	–	–			1 506	1 111	
3½"		–		–	1 620	985	–	1 350				1 396						–	–	1 303	985
8" Unidade	–		828	923														892	1 507	972	
Dos lados			984	1 064	1 020													1 068		1 442	

Estos precios corresponden a los vigentes al día 12 de noviembre de 1985
corresponden a precios unitarios y con el IVA incluido

Los estudios, cotizaciones y otras investigaciones que realiza SERVIPRES pueden solicitarse al fono 715852. También se puede pedir la visita de un representante o suscribirse a sus servicios informativos permanentes.

Inician el SINACOFI

De un procesamiento mínimo de 1 200 mensajes por hora será capaz el Sistema Nacional de Comunicaciones Financieras (SINACOFI) cuando entre en funciones. Los contratos para la elaboración del sistema fueron firmados en noviembre último entre la Asociación de Bancos e Instituciones Financieras de Chile A.G. y las empresas IBM y Dicom.

En la construcción del SINACOFI se utilizan modernas tecnologías que incluyen un sistema de respaldo automático para el caso de fallas de uno de los computadores centrales, un sistema mediante el cual la información que transita por la red es ininteligible, usando códigos secretos y sistemas de claves que restringen el acceso a la red, según las distintas funciones y montos de las operaciones.

Chile es el primer país latinoamericano que tiene un sistema de este tipo, el que debe entrar en funciones a fines del próximo año.

Otras de sus virtudes son su tiempo de respuesta inferior a ocho segundos y una capacidad para mantener almacenados en línea los últimos 25 mil mensajes.

En la confección del sistema IBM proporcionará el hardware y Dicom el software.

ENERA de aniversario

Un año de existencia cumplió el 29 de noviembre recién pasado la Escuela de Negocios de Rancagua (ENERA) que en su calidad de centro de formación técnica ofrece cursos y seminarios sobre computación.

Entre las actividades programadas con este motivo se cuenta la realización de la Primera Muestra de Microcomputación de la Sexta Región a desarrollarse en las ciudades de Rancagua y San Fernando.

Cabe destacar que ENERA auspicia el Club de Microcomputación de Rancagua, uno de los más numerosos y entusiastas del país.

Siguen negociaciones para venta de ECOM

Trecientos sesenta millones de pasos cobra Corfo a los trabajadores de ECOM por la venta de esta empresa estatal de computación El traspaso incluye equipamiento y bienes inmuebles entre los que se cuenta el modemo edificio de Lo Curro, actualmente arrendado a la FAO

Cuando el Consejo de Ministros de Estado decidio la venta del 100 por ciento de las acciones de ECOM, la asamblea de trabajadores determinó defender su permanencia como empresa estatal

El acuerdo obedecia, segun Augusto Leighton, miembro de la comisión, al convencimiento de que es imprescindible para el Estado contar con una empresa de computación e informática con un rol muy bien establecido, que, además, no necesita competir con sus similares del área privada

Como esta primera proposición la rechazó el Gobierno, los trabajadores, a través de su Sindicato, ofrecieron comprar ECOM, lo que fue aceptado

"El interés de comprar es consecuencia de la necesidad de asegurar la continuidad laboral del máximo de personal", puntualizó Leighton La operación involucra a 110 personas, que constituyen el 95% de los trabajadores de ECOM

ASICOM lanza computador en FISA

Con una charla explicativa y demostraciones prácticas de su aplicación, se lanzó al mercado —en el restorán Munchen de FISA— el nuevo computador DIMENSION

Con tecnología de multiprocesamiento, este equipo presenta las caracteristicas que antes existian sólo para los grandes computadores doce estaciones de trabajo inteligentes, el concepto de memoria Ram Caché, capacidad de conectarse en red hasta con 50 PC y compatibilidad en hardware y software con IBM PC XT

Con la posibilidad de adaptarse a programas anteriores de otros equipos el software nacional del DIMENSION incluye sistemas administrativos desarrollados por ASICOM tales como contabilidad con cuentas corrientes, control de stock con multibodegas y remuneraciones

Software para la agricultura

Dos programas aplicables en la agricultura, se encuentra ofreciendo a los interesados la empresa ASC, distribuidora en Chile de los computadores Hewlett Packard Uno es el Sistema de Administración de Lecheria (SAL) y el otro es un Sistema de Formulacion de Alimentos mediante Raciones de Minimo Costo (FARM) util para los productores agrícolas que necesitan calcular exactamente los componentes nutritivos de la alimentación del ganado, conejos, aves y otros

Por otra parte ASC también tiene novedades para el comercio detallista Para este sector dispone de un sistema de venta al detalle, cuya principal caracteristica es que el control de venta se realiza a través de un lector de codigo de barras

El SAL y el FARM, adquiridos en paquete, tienen un valor de 1 500 dólares Por separado el primero cuesta US$ 890 y el FARM US$ 780

Muy requerido al QL de Sinclair

Gran demanda tuvo en FISA el QL, computador de 32 bits de Sinclair, con 128 K en RAM expandibles a 640 Ejecutivos de esa empresa señalaron que el producto se agoto debiendo tomar pedidos para entrega a 30 dias El valor del QL es de 115 mil pesos mas IVA

También anunció Sinclair que para el ano 1986 nombrará un numero limitado de distribuidores a través de Chile (sólo siete en Santiago)

VI Congreso de Ingeniería Eléctrica

La humanidad está experimentando el proceso evolutivo que se ha llamado Segunda Revolucion Industrial que es una consecuencia de los microcomputadores y que constituye un proceso social del cual deben esperarse aun consecuencias más profundas que las que tuvo la Primera Revolución Industrial



Esta afirmación la hizo Igor Saavedra de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile en la conferencia inaugural –que tuvo a su cargo– del VI Congreso Chileno de Ingenieria Electrica realizado en noviembre recién pasado y organizado por el Departamento de Ingeniería Eléctrica y Ciencia de la Computación de la Universidad Católica

En el evento fue posible apreciar un aumento en la cantidad y calidad de los trabajos con respecto a los anteriores Congresos que vienen desarrollándose desde 1975, cada dos años En esta ocasión se recibieron 172 trabajos provenientes de Chile, Brasil, Argentina Yugoslavia Egipto, Singapur, Canadá y Estados Unidos De ellos 109 fueron aceptados por el Comite Editorial para su presentación en el Congreso

Inventar un camino propio de desarrollo es una responsabilidad muy definida para los intelectuales y en particular para los científicos y tecnólogos del pais Para que esta posibilidad pueda hacerse concreta, es necesario desde luego que Chile tenga ciencia y tecnología de un nivel apropiado y esto implica un esfuerzo nacional que todavía tiene que realizarse Por otra parte el pais necesita para su propio futuro contar con personas capaces de imaginar instrumentos y estilos novedosos y apropiados a las necesidades chilenas Estas fueron otras de las conclusiones senaladas por Igor Saavedra en su alocución

El presidente del Congreso, Hugh Rudnick Van De Wyngard destacó a MICROBYTE el gran numero y la calidad de los trabajos que se refirieron a aplicaciones computacionales Puso de relieve la presencia del doctor ingeniero Plinio Stange de Sao Paulo, Brasil quien intervino dictando un curso de perfeccionamiento titulado 'Introducción a la automatización de los sistemas de producción

Presencia de SICOB en Chile

Con la participacion de Max Hermieu, Presidente de Sicob, se realizó a principios de noviembre una conferencia en la que se informó respecto a la trascendencia de este importante evento informático que se realiza año a ano en Francia

El Sicob Salón Internacional de Informática, Telemática Comunicaciones y Burótica nacio n 1949 y cada año, en abril y eptiembre presenta en Paris la ultima tecnología en gestion, equipamiento y organización de las empresas

En 1985, Sicob fue visitado por más de medio millón de personas especialistas provenientes de 115 países Expositores fueron 1238 empresas que presentaron productos afines al área informatica

Paralelamente a Sicob se realiza anualmente también una Convención Informática en la que especialistas disertan sobre distintos tópicos, sirviendo además como punto de encuentro para profesionales de diversos paises Este ano entre los expositores se contó a Bill Gates Presidente de Microsoft e Yves Colson de American Motors



M[illegible]
Pr[illegible] o General SICOB

Software para el Estado

La empresa de computación Conde creó para las entidades estatales, bajo las normas impartidas por la Contraloría General de la Republica, un software de contabilidad general de la nacion

El programa que es totalmente interactivo, está escrito en Cobol ANS y puede correr en IBM PC, IBM sistema 34 36 38 4300 y 3000 XX

El nuevo sistema integra la contabilidad patrimonial y la presupuestaria de los organismos del Estado Antes la Contraloría utilizaba sólo sistemas de contabilidad presupuestaria que eran construidos bajo modalidad BATCH

latindata:

CLARIDAD CENTRONICS

"EL FABRICANTE MUNDIAL MAS IMPORTANTE EN IMPRESORAS"



En materia de impresoras, decir CENTRONICS es señalar siempre lo mejor

CENTRONICS ofrece toda una gama de equipos confiables que van desde las eficientes G L P de 50 carácteres por segundo las HORIZON de 160, pasando por las LW, que imprimen 400 y 800 LPM hasta la sofisticada SERIE E, que imprime hasta 2 400 líneas por minuto.

Además, todas ellas imprimen con calidad carta.

Es decir, hay una respuesta CENTRONICS para cualquier necesidad específica que su empresa tenga que solucionar.

Y CENTRONICS, la mejor impresora que se fabrica, está tambien en LATINDATA.

VENGA por su CENTRONICS a LATINDATA.



Eliodoro Yanez 2596
Telefonos 460205 42209
Nueva York 68
Telefonos 6980479 723412
Santiago

PERKIN-ELMER * ONTEL-VISUAL * CALCOMP * CENTRONICS

Selección de proyectos de informática

Guillermo Beuchat S.
Ing. Civil Industrial U. de Chile

La mayoría de las empresas modernas, conscientes de la necesidad de introducir sistemas de información computarizados en su gestión, se enfrentan a un dilema clásico: ¿qué sistemas desarrollar, y en qué orden? La selección de proyectos adecuados es indispensable para tener éxito al introducir la herramienta computacional, pues se corre el riesgo de incurrir en altos costos y no alcanzar los objetivos propuestos. Lo anterior es especialmente relevante en el caso de empresas sin experiencia en el procesamiento electrónico de datos, aunque la selección adecuada de proyectos también es importante al ampliar la cobertura de los sistemas en organizaciones que ya cuentan con la tecnología en distintas áreas, y en el que existe un departamento de informática formalmente definido.

Este trabajo presenta una metodología para la identificación, evaluación y selección de proyectos de informática, que contempla mecanismos adecuados para lograr cada uno de estos objetivos en forma estructurada. Dada la extensión limitada de este artículo, solamente se describen en forma resumida las diferentes etapas de la metodología propuesta, advirtiéndose la necesidad de profundizar y ampliar cada una de ellas.

La necesidad de planificar

El desarrollo de un plan de informática para la empresa, como paso previo a la definición de los sistemas específicos que se desea implementar, resulta indispensable para lograr una adecuada planificación del proceso de computarización de la empresa. Existen varios factores claves en los sistemas de información computarizados, entre ellos el tamaño y la complejidad que hacen conveniente un análisis exhaustivo de requerimientos y tiempo esperado de desarrolló antes de iniciar el diseño. En una discusión acerca del tamaño óptimo que debiera tener un sistema, desde el punto de vista de la capacidad de la empresa para desarrollarlo, DONALDSON (1) propone el siguiente:

"no mucho mayor que el último proyecto que completamos con éxito"

Si aplicamos este axioma a una empresa que recién incorpora la tecnología computacional, llegamos a la conclusión de que es conveniente partir por desarrollar sistemas pequeños, con objetivos claramente definidos, fáciles de administrar y con muy buenas perspectivas de éxito. Una planificación adecuada es entonces un factor clave en el éxito o fracaso del desarrollo o implementación de sistemas en una empresa.

La figura 1 muestra la metodología de selección de sistemas de información que se propone para contribuir a la solución del problema planteado, basada en un trabajo similar realizado por BROOKS (2). Las diferentes etapas de la metodología se analizan separadamente a continuación, en una forma que permite su implementación directa por los ejecutivos del área.

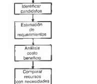


Figura 1

Fijación de objetivos y políticas

Corresponde a la administración superior de la empresa la determinación de los objetivos y políticas de desarrollo de la actividad de procesamiento de datos. Tal como lo sostiene ROCKART (3), los ejecutivos máximos de la organización deben definir claramente cuáles son sus necesidades de información, de tal forma que puedan asignarse prioridades y objetivos de corto, mediano y largo plazo.

En cuanto a los objetivos que es necesario plantear, los siguientes son especialmente relevantes:

"desarrollar en primer lugar sistemas "estratégicos" que permitan a la empresa competir en mejor

forma en el mercado La definición de sistemas estratégicos en una empresa es un tema que ha sido abordado anteriormente en MICROBYTE (4)

*construir sistemas de apoyo a la gestion y no simples mecanizaciones de procesos manuales, aun cuando por ello se obtengan beneficios tangibles

*desarrollar todos los sistemas a mínimo costo y máximo rendimiento

*maximizar la compatibilidad e interacción entre los sistemas

*plantear y mantener estándares de auditoria y seguridad comunes a todos los sistemas

*minimizar el impacto negativo de la incorporación de la tecnologia en la empresa

*maximizar el retorno tangible e intangible de cada sistema que se desarrolle

*obtener la colaboración permanente de los usuarios en el diseño de los sistemas

El conjunto de proyectos que se escoja para desarrollar en un periodo determinado deberia cumplir al menos estos objetivos sin perjuicio de otros que puedan plantearse de acuerdo al tipo de negocio o empresa En cuanto a las politicas generales de la empresa, se puede mencionar las siguientes

*definición de una linea tecnologica clara, comprometiendo las adquisiciones de hardware y software con productos compatibles 'hacia arriba' que permitan una expansión posterior

*definicion de politicas de contratación y uso de personal propio en proyectos del area computacional

*organización formal de la actividad de procesamiento de datos y desarrollo de sistemas, y su relacion con el organigrama general de la empresa

*definición de metas de largo plazo en cuanto al nivel de importancia de los sistemas para la gestión de la empresa

*definición de areas estrategicas del negocio en las cuales seria deseable el uso de la tecnologia a fin de desarrollar una ventaja comparativa

La fijación de objetivos y politicas de informática a nivel de toda la empresa es el primer paso para determinar una cartera de proyectos a desarrollar, pues todos ellos deben encuadrarse dentro de estas lineas generales

Identificación de candidatos

Antes de seleccionar los proyectos que se van a implementar, es necesario identificar todos los sistemas que tendrian utilidad para la empresa, o permitirian una mejor gestión a través de la información que proporcionan Estos serán los candidatos entre los cuales será necesario priorizar y seleccionar los sistemas que se implementaran finalmente

Existen dos enfoques posibles para realizar la identificacion de candidatos un esquema 'top-down', en que se analizan los aspectos claves de la organización desde los niveles administrativos superiores hacia abajo, y otro evolutivo o "bottom-up" que parte analizando deficiencias en los sistemas actuales Este ultimo enfoque implica el peligro de desarrollar sólo soluciones de corto plazo ignorando las politicas globales establecidas en la etapa anterior

El enfoque "top-down", por otra parte, permite identificar sistemas claves sin perder de vista los objetivos de la empresa, a través de las siguientes actividades

*análisis de las diferentes áreas de la empresa, en conjunto con los ejecutivos responsables de cada una Se busca identificar las decisiones que se toman que información utilizan para tomarlas y cómo la obtienen El uso de técnicas de análisis estructurado (Diagramas de flujo Warnier-Orr) puede ser util en esta fase de la identificacion Se debe enfatizar y profundizar el análisis en aquellas áreas que han sido calificadas como estrategicas

*estudio de las restricciones que enfrentan las áreas estratégicas y que limitan su rendimiento Por ejemplo, se puede analizar el tiempo requerido para efectuar busquedas en archivos, disponibilidad de capital de trabajo, malas relaciones laborales, mala imagen con los clientes, incapacidad para mejorar las ventas, etc

*analisis de estas restricciones, a fin de determinar cómo un sistema computacional podria levantarlas o mejorar el rendimiento Para ello se debe considerar las caracteristicas propias de cualquier sistema computacional gran velocidad de proceso, gran capacidad de almacenaje posibilidad de efectuar consultas interactivas, facilidad para efectuar el seguimiento de niveles de stock o cantidades de dinero etc

*proposición de sistemas computarizados especificos, que ofrezcan solución a los problemas detectados

*analisis de las proposiciones a la luz de las politicas y objetivos globales de la empresa a fin de determinar si los sistemas propuestos se encuadran o no en ellas

Estimación de requerimientos

Tras la identificación de los sistemas candidatos, es necesario efectuar una estimación de los recursos necesarios para su implementación En primer lugar, definamos las variables que debemos estimar

*el tiempo necesario para completar cada actividad o trabajo dentro del proyecto

*el tiempo necesario para completar todo el proyecto Este tiempo no tiene por qué ser la suma de los tiempos asignados a cada actividad, pues muchas de éstas podrian realizarse en forma simultánea

*recursos necesarios, tanto humanos como en tiempo de computación o uso de hardware

*costo de remuneraciones y equipos

El problema de la estimación es la naturaleza esencialmente variable de la especificación o diseño de un sistema y las necesidades de recursos durante el desarrollo Normalmente, la duración de una actividad e incluso su existencia misma dentro del proyecto puede verse afectada. Por cambios en las necesidades de los usuarios o en la estructura jerárquica de la empresa Por ello,

los métodos de estimación deben considerar la probabilidad de cambio, ya sea de una revisión permanente o el uso de factores de holgura

A continuación, se presentan algunas técnicas de estimación que puedan usarse para cuantificar las variables que hemos definido

*Apreciación Subjetiva, el analista adivina la magnitud de cada variable basado en una apreciación subjetiva de la complejidad de ella Dado que no es un enfoque formal el estimador normalmente estará influenciado por presiones o actitudes de quienes deben autorizar el proyecto lo que conduce a una subestimación de estas magnitudes en la generalidad de los casos

*Descomposición y Síntesis – el analista divide el proyecto en pequeños módulos de trabajo, cuyas variables pueden estimarse con relativa precisión en base a su experiencia personal Luego se suman todas estas estimaciones para obtener la estimación de cada variable a nivel de proyecto completo

*Análisis histórico – si existen datos históricos acerca de las variables en proyectos similares, pueden usarse estos valores en el Proceso de Descomposición y Síntesis, en lugar de estimaciones subjetivas basadas en la experiencia del analista

*Análisis paramétrico – consiste en analizar estadísticamente las variables de tiempo recursos y costo en una gran cantidad de sistemas diferentes, a fin de obtener ecuaciones de regresión múltiple que relacionen estas variables Aunque estos métodos no tienen aceptación general, entregan órdenes de magnitud bastante aceptable sobre los cuales basar la estimación POPE (5), en un estudio realizado en el ejército de los Estados Unidos sobre un total de 20 sistemas computacionales obtuvo entre otras las siguientes ecuaciones que se presentan a modo de ejemplo

$A = 2.57B = 5.1C + 0.12D$
$E = 0.38F + 0.19G$

en que A = meses/hombre totales del sistema
B = número de formatos de salida
C = número de registros diferentes
D = número de transacciones de input por mes (miles)
E = número horas computador por mes (operación)
F = tipos de transacciones diferentes
G = número de campos de entrada de datos

Antes de concluir nuestro análisis de la estimación de requerimientos para un proyecto, es necesario destacar algunos aspectos importantes, propuestos por BROOKS (6)

*el tiempo y el recurso humano no son intercambiables Si seis personas demoran un año en completar una actividad tres personas puedan demorar bastante más de dos años

*suponer personas de rendimiento "promedio" y luego ponderar por un factor cuando se asigne la persona a un trabajo específico

*agregar suficiente tiempo para consultar documentación, revisión de errores y enseñanza de subalternos a los analistas de un proyecto Algunos estudios indican que solamente el 60% del tiempo total asignado a un analista estará dedicado al diseño del proyecto y el resto se empleará en labores anexas

Análisis de costo/beneficio

A fin de seleccionar los proyectos, es necesario efectuar una estimación de los costos y beneficios de cada uno, en términos monetarios Para ello usaremos las estimaciones de recursos calculadas en la etapa anterior, valorándolos de acuerdo a un criterio uniforme para todos los proyectos y a los precios reales relevantes para la empresa Esto implica, por ejemplo, que el costo de una hora de analista no corresponde necesariamente al sueldo que se le paga, sino el rendimiento que podría tener esa misma hora de trabajo en un proyecto alternativo de mayor rentabilidad Aplicaremos entonces el criterio económico de costo de oportunidad en la valoración de todos los recursos que utiliza el sistema ya sean insumos, recursos humanos o uso de tiempo de computación

Los costos asociados al desarrollo de un sistema son, entre otros, los siguientes

*costos asociados al personal ya sea propio contratado o en consulta
*costos asociados al uso de computador
*costos asociados al proceso de puesta en marcha y operación paralela
*costos de adquisición de hardware y software
*costos de mantención de hardware y software
*costos tributarios por depreciación de equipos
*costos de materiales e insumos de operación
*costos de capacitación de los usuarios
*costos "intangibles" derivados del uso del sistema, ya sea en relaciones humanas fallas del software o falta de flexibilidad de los procesos administrativos asociados al sistema

La identificación y cuantificación de los costos siempre resulta más fácil que cuantificar los beneficios, pues estos son en su mayoría intangibles



Continua en pag 20

La reducción de personal a causa de la implementación de los sistemas computacionales es un mito pues lo que ocurre en realidad es un cambio sustancial en el tipo de trabajo realizado con un aumento de la productividad de la empresa sin aumentar el personal

Entre otros, podemos mencionar los siguientes beneficios de un sistema computacional, sin perjuicio de beneficios propios de sistemas específicos

*disminución del capital inmovilizado en inventarios
*mejoras en el control de calidad y disminución en el numero de rechazos
*mejor uso de recursos e insumos
*disminución del numero de errores que requieren reproceso
*mejoras en la planificación de la producción
*ahorros tributarios
*economías de escala
*mejoras en el ambiente laboral
*mejoras en la imagen de la empresa
*mejoras en la gestión de la empresa
*disminución en el tiempo de entrega de productos
*mejoras en el control de activos

Debe hacerse un esfuerzo por cuantificar estos beneficios incluso a través de métodos indirectos, a fin de obtener perfiles monetarios para los proyectos que se desea priorizar Por otra parte, la priorización económica debe hacerse sobre proyectos de la misma "vida util", lo que podría implicar modificaciones o ajustes a los flujos de caja en esta etapa

Comparación de recursos con necesidades

El siguiente paso en el proceso de selección de proyectos es descartar aquellos infactibles desde el punto de vista de la disponibilidad de recursos, ya sea monetarios, computacionales o humanos Para ello se analizan los siguientes factores

*nivel de inversión inicial
*costos de operación rutinaria
*disponibilidad de personal técnico calificado
*requerimientos administrativos y organizacionales
*capacidad de absorción tecnológica de la empresa
*restricciones legales

El resultado de esta comparación deberá ser una cartera de proyectos factibles, que es necesario priorizar para proceder a su implementación. El análisis de recursos de esta etapa debe hacerse como si cada proyecto fuera el único, es decir como si todos los recursos de la empresa se usaran en su implementación solamente. De esta manera se obtiene una cartera de proyectos individualmente factibles, tal que la suma de recursos necesarios para implementarlos supera probablemente la capacidad de la empresa La ultima etapa de esta metodología se encargará entonces de entregar una secuencia de implementación que deberá seguirse hasta agotar los recursos disponibles

Priorización de proyectos

La priorización de la cartera de proyectos se realiza mediante una evaluación financiera con indicadores adecuados, mediante un analisis de factores intangibles dependientes de las políticas y objetivos de la empresa, o mediante una combinación adecuada de ambos enfoques. Corresponderá entonces a la administración superior de la empresa decidir cuál de estos criterios usa para priorizar y ordenar la cartera de proyectos

Dado que los métodos de evaluación financiera de proyectos han sido ampliamente tratados en diversas publicaciones, incluso en MICROBYTE (7), nos limitaremos a mencionar aquí algunos indicadores aplicables al problema que nos ocupa.

*Tiempo de recuperación del capital – se define como el tiempo necesario para que los flujos de caja positivos generados por el nuevo sistema equiparen la inversión y los costos de operación

*Punto de equilibrio – se define como el momento en que los flujos de caja positivos del nuevo sistema se equiparan con los del sistema actual La figura 2 ilustra el concepto de punto de equilibrio en el contexto de nuestro problema

*Valor presente neto – se define como la suma de los flujos netos de caja del nuevo sistema, descontados a la tasa de actualización relevante para la empresa

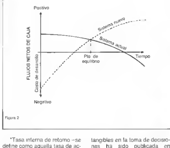


Figura 2

*Tasa interna de retorno —se define como aquella tasa de actualización que genera un valor presente neto igual a cero y corresponde a la rentabilidad real del proyecto

Por otra parte el análisis de los factores intangibles relacionados con las politicas globales de la empresa puede hacerse a través de un método de asignación de puntajes Dicho metodo calcula un puntaje total mediante el uso de ponderadores para cada factor y puntajes parciales para cada factor en cada proyecto Existen dos enfoques para calcular estos puntajes

*Enfoque aditivo —se calcula un puntaje global para el proyecto i de acuerdo a la siguiente formula

$$P_i = \sum_{j=1}^{n} P_{ij} * W_j$$

en que P_i es el puntaje total del proyecto i P_{ij} es el puntaje del factor j en el proyecto i W_j es la ponderación del factor j en la decisión total, y n es el numero de factores en consideración

*Enfoque multiplicativo — se calcula un puntaje global para el proyecto i de acuerdo a la siguiente fórmula

$$P_i = \prod_{j=1}^{n} P_{ij}^{W_j}$$

usando la misma nomenclatura del enfoque anterior

Una discusion más detallada de este método para evaluar intangibles en la toma de decisiones ha sido publicada en MICROBYTE (8)

Referencias

(1) A GUIDE TO THE SUCCESSFULL MANAGEMENT OF COMPUTER PROJECTS
Hamish Donaldson
Associated Bussiness Press 1978

(2) INFORMATION SYSTEMS DESIGN
Frederick P Brooks
Prentice-Hall, Inc 1982

(3) CHIEF EXECUTIVES DEFINE THEIR OWN DATA NEEDS
John F Rockart
Harvard Bussiness Review March-April 1979

(4) LOS SISTEMAS DE INFORMACION COMO ARMAS ESTRATEGICAS DE LAS EMPRESAS
Guillermo Beuchat S
MICROBYTE agosto 1985

(5) ESTIMATING THE RESOURCES NEEDED FOR EDP SYSTEMS
Dorothy J Pope
Economics of Information
North Holland 1975

(6) THE MYTHICAL MAN-MONTH
Frederick P Brooks
Addison-Wesley 1975

(7) EVALUACION DE PROYECTOS
Guillermo Beuchat S
MICROBYTE enero 1985

(8) PROBLEMAS DE LOCALIZACION DE PLANTAS Y/O BODEGAS
Guillermo Beuchat S
MICROBYTE julio 1985

Proyecto camino

HDRACIO VALDES BUNSTER
ING. EJEC. DE MINAS

Los caminos son tan viejos como el hombre Las necesidades cotidianas de los antiguos grupos humanos, como por ejemplo buscar agua, cazar y otras tareas domesticas, crearon en forma natural los caminos Incluso antes de inventarse la rueda ya existian los caminos La apanción de la rueda movilizó todo un progreso que se traduce después de 5 000 anos en la actual civilizacion tecnologica Las necesidades han cambiado los pequeños grupos humanos que antes trazaban caminos espontaneamente se han transformado en grandes aglomeraciones urbanas Ya los caminos responden a necesidades de eficiencia para satisfacer las exigentes necesidades del mundo actual Es por eso que dentro de la ciencia de Ingenieria tanto las lineas ferreas los canales de regadio y naturalmente los caminos deben tener inclinaciones poco pronunciadas

En el aspecto económico una de las maneras de minimizar los costos en el trabajo es considerar que el volumen de material que se debe remover sea cercano o igual al volumen de material que se debe colocar lo que al mismo tiempo resuelve el problema de la inclinación de la obra

Un caso real

Imaginemos que se desea construir un camino entre dos ciudades cualesquiera Lo primero que se debe realizar es estudiar el terreno y segun esto decidir cual sera la trayectoria a seguir asi luego un equipo Topográfico efectuara mediciones en terreno, con los cuales despues de un sencillo calculo, se obtienen unas alturas respecto al nivel del mar las que llamaremos cota (C), (la cota del nivel del mar es de referencia e igual a cero)

Los puntos de medida deberán ser criteriosamente elegidos y coincidir con el eje longitudinal del camino de forma que representen la realidad Topografica Ademas, deben medirse las distancias horizontales entre cada punto donde se midieron cotas (*)

Una vez finalizado el trabajo de terreno se procesan los datos de forma tal de obtener una inclinacion del camino que cumpla con las premisas antes mencionadas y es precisamente esta la tarea que el programa adjunto realiza basado en una serie de relaciones geometricas y considerando una toma lineal de puntos de muestreo

Inicialmente el programa da la opción al usuario de proponer su pendiente o en caso contrario el programa buscara una pendiente y por retroalimentacion de datos la ajustará hasta encontrar la optima Para ambos casos primeramente calcula la cota rasante (CR) que como su nombre lo indica es la proyeccion de la cota de terreno sobre la pendiente

(*) Todos los datos necesarios para el cálculo pueden ser obtenidos en terreno por medio de un Taquímetro

El valor de la cota rasante para cada punto se obtiene por medio de la relación

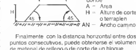
$$CR_i = CR_{i-1} + (P * D_{(i,i-1)}) / 100$$



C = Cota del terreno
CR = Cota rasante
P = Pendiente (%)
D = Distancia horizontal entre puntos

Conociendo el valor de la cota rasante de un determinado punto y por supuesto la cota de terreno del mismo se puede decidir por medio de una diferencia si en dicho punto se tendra que excavar o rellenar el terreno

Ahora bien, considerando un ancho de camino igual a AN y un talud natural del material de relleno o de corte de 45°, puede obtenerse el área transversal de relleno o de corte



con

A = H * (AN + H)

donde

A = Area
H = Altura de corte o terraplen
AN = Ancho camino

Finalmente con la distancia horizontal entre dos puntos consecutivos, puede obtenerse el volumen de material de relleno o de corte de un bloque

Este programa acepta N puntos de medición con N= 30 (si se requiere mayor capacidad basta aumentar las constantes en la linea 95), y por tanto calculara el volumen de N-1 bloques Para esto el programa divide el trayecto a realizar como muestra la fig

HC = Altura de corte
SC = Area de corte
HT = Altura de relleno
ST = Area de relleno
D = Distancia horizontal
P = Pendiente (%)



Continua en pag 24



Como se observa en la figura anterior, existen 4 alternativas respecto a la naturaleza del volumen de los bloques, estas son: que las dos areas transversales que limitan al bloque sean de relleno, que ambas sean de corte o que la primera sea de relleno y la segunda de corte, o viceversa.

La manera que el programa calcula los volumenes de material por bloque, esta basado en las siguientes relaciones, segun sea el caso:

a) Si ambas secciones son de relleno, el cálculo de volumen es como sigue:

$$VT_{(i,\ i+1)} = [(ST_i + ST_{i+1})/2] * D_{(i,\ i+1)}$$

VT = Volumen de terraplén



b) Analogamente, si ambas secciones son de corte, el volumen de corte (VC) se calcula como sigue:

$$VC_{(i,\ i+1)} = [(SC_i + SC_{i+1})/2] * D_{(i,\ i+1)}$$



c) Finalmente si una sección es de corte y la otra de terraplen, o viceversa, el volumen de corte y de terraplen respectivo se calculan como sigue:

$$VT_{(i,\ i+1)} = \frac{ST_i}{2} * DT$$

$$DT = [ST_i / (ST_i + SC_{i+1})] * D_{(i,\ i+1)}$$

$$VC_{(i,\ i+1)} = \frac{SC_{i+1} \cdot DC}{2}$$

$$DC = [S_{i+1} / (SC_{i+1} + ST_i)] \cdot D_{(i,\ i+1)}$$



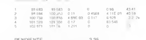
Uso del programa

Se alimento el programa a modo de ejemplo con los siguientes datos en un COMMODORE 64

Numero de puntos ? 5
Ancho camino ? 9
Cota del punto 1 ? 99 683
Cota del punto 2 ? 99 894
Distancia entre puntos 1 y 2 ? 20
Cota del punto 3 ? 100 738
Distancia entre puntos 2 y 3 ? 15
Cota del punto 4 ? 101 528
Distancia entre puntos 3 y 4 ? 15
Cota del punto 5 ? 102 971
Distancia entre puntos 4 y 5 ? 12
Fija la pendiente s n ? N

**** ESPERE ****

y se entregan los siguientes resultados



PTO	COTA	CR	HC	HT	VC	VT
1	99 683	99 683	0	0	0 98	43 41
2	99 894	100 353	0 11	0 4568	4 11E 03	40 59
3	100 738	100 856	4 89E 03	0 117	6 928	3 2 2
4	101 528	101 350	0 17	0	83 548	
5	102 971	[illegible]	1 211	0	0	0

PENDIENTE 3 35
VOLUMEN DE CORTE 91 4714493
VOLUMEN DE TERRAPLEN = 87 2868573
DIFERENCIA DE VOLUMEN = 4 18459198

Cabe mencionar que si se desean valores mas pequenos de diferencia de volumen basta con disminuir los valores (constantes) de la iteracion en las lineas 810-840 del programa, sin embargo por tratarse de valores muy pequeños de incremento el tiempo de ejecucion aumenta en gran proporcion.

Continua en pag. 26 Programa

```
10 REM** PROYECTO CAMINO **
20 REM ** HORACIO VALDES B. **
30 VC=0 DV=0 TT=0 PX=0 PT=0 W=1
35 PRINTCHR$(147)
40 INPUT"NUMERO DE ESTACIONES";ES
45 DIM C(20),D(20),CR(20),H(20),HC(20),HT(20),VC(30),VT(30),SC(30),ST(20)
100 INPUT"ANCHO CAMINO";AN
110 FOR I=1 TO ES
120 PRINT"COTA DEL PUNTO";I: INPUT C(I)
130 IF I>=2 THEN PRINT"DISTANCIA ENTRE PUNTO";I-1;"Y";I: INPUTD(I)
140 NEXT I
145 INPUT"FIJA LA PENDIENTE S/N";A$
150 IF A$="S" THEN INPUT"SU PENDIENTE";P:GOTO 170
155 IF A$<>"N" THEN GOTO 145
157 IF A$="N" THEN P=0:PRINT:PRINT:PRINTTAB(9);"**** ESPERE ****"
170 REM ** CALCULO COTA RASANTE **
175 C=C+1
180 FOR H=1 TO ES
190 CR(H)=C(H)
200 IF H>1 THEN CR(H)=CR(H-1)+(P*D(H))/100
210 NEXT H
220 REM *** HC Y HT ***
230 FOR I=1 TO ES
240 H(I)=C(I)-CR(I)
250 IF H(I)>=0 THEN SC(I)=H(I)*(AN+H(I)):HC(I)=H(I):ST(I)=0:GOTO 270
260 IF H(I)<0 THEN HT(I)=-H(I):ST(I)=HT(I)*(AN+HT(I)):SC(I)=0
270 NEXT I
280 REM ** CALCULO DE VOLUMEN **
290 FOR K=1 TO ES
300 IF ST(K)=0 AND ST(K+1)=0 THEN VC(K)=((SC(K)+SC(K+1))/2)*D(K+1)
310 IF SC(K)=0 AND SC(K+1)=0 THEN VT(K)=((ST(K)+ST(K+1))/2)*D(K+1)
320 IF SC(K)>0 AND ST(K+1)>0 THEN GOSUB 500
330 IF ST(K)>0 AND SC(K+1)>0 THEN GOSUB 550
340 NEXT K
350 REM **CALCULA VC Y VT**
360 VT=0:VC=0:DV=0
370 FOR I=1 TO ES-1
380 VT=VT+VT(I):VC=VC+VC(I)
390 NEXT I
400 DV=VC-VT
410 IF PX=1 OR PT=1 THEN GOSUB 600
420 IF C=1 AND A$="S" THEN GOSUB 600
430 IF A$="N" THEN GOSUB 800
440 IF CV=1 THEN GOSUB 600
450 GOTO 170
500 REM ** CORTE TERRAPLEN **
510 DC=(SC(K)/(SC(K)+ST(K+1)))*D(K+1):VC(K)=(SC(K)/2)*DC:VC=VC+VC(K)
520 DT=(ST(K+1)/(ST(K+1)+SC(K)))*D(K+1):VT(K)=(ST(K+1)/2)*DT:VT=VT+VT(K)
530 RETURN
550 REM ** CORTE TERRAPLEN **
560 DC=(SC(K+1)/(SC(K+1)+ST(K)))*D(K+1):VC(K)=(SC(K+1)/2)*DC:VC=VC+VC(K)
570 DT=(ST(K)/(ST(K)+SC(K+1)))*D(K+1):VT(K)=(ST(K)/2)*DT:VT=VT+VT(K)
580 RETURN
600 REM ** IMPRESION **
605 PRINTCHR$(147)
610 PRINT"PTO   COTA   CR    HC   HT   VC   VT"
615 PRINT"******************************************"
616 PRINT
620 FOR I=1 TO ES
630 PRINTI,TAB(6),C(I),TAB(13),CR(I),TAB(19),HC(I),TAB(24),HT(I),TAB(29),VC(I)
T I
635 PRINT
636 IF I=4*W THEN GOSUB 1000
640 NEXT I
645 PRINT
650 PRINT"PENDIENTE               =";P
660 PRINT"VOLUMEN DE CORTE        =";VC
670 PRINT"VOLUMEN DE TERRAPLEN    =";VT
680 PRINT"DIFERENCIA VOLUMEN      =";DV
690 END
700 RETURN
800 REM ** ITERACION **
805 IF DV=0 THEN CV=1
810 IF DV>0 AND TT=0 THEN P=P+0.1:CC=1
820 IF DV<0 AND CC=0 THEN P=P-0.1:TT=1
830 IF CC=1 AND DV<0 THEN PX=1:P=P-0.05
840 IF TT=1 AND DV>0 THEN PT=1:P=P+0.05
850 RETURN
1000 REM ** PAUSA RESULTADOS **
1010 PRINT:INPUT "PARA CONTINUAR PRESIONE RETURN ";B$:PRINTCHR$(147)
1020 W=W+1
1030 RETURN
READY.
```

SECCION x MARCAS

VECTOR

Desglose de dineros

Roberto Claternas Aliste

Una actividad que no por ser comun deja de ser molesta, es cuando a fin de mes en una empresa hay que ir al banco y retirar en efectivo los sueldos para el personal

El problema que se presenta, es que no sólo debemos retirar la suma de los sueldos, sino además el dinero debe venir desglosado de tal manera que a cada uno se le pueda cancelar la cantidad correspondiente sin necesidad de correr a la botillería del frente para que nos cambien un billete de cinco mil

Este programa, hecho en un Vector-4, comienza preguntando por el numero de sueldos a pagar y cuánto corresponde a cada uno Para traspasarlo a otro computador, con un dialecto Basic distinto, basta saber que la Instruccion FNAT$ es una función que sirve sólo para tabular en la pantalla (linea 90) y el CHR$(4) en la linea 10, borra la pantalla

De hecho, las lineas modulares del programa, donde se encuentra el algoritmo de desglose, son desde la 170 a la 390

Como resultado, el programa entrega la sumatoria de los sueldos y un cuadro con la cantidad de billetes y monedas de cada tipo que hay que retirar.

```
INDIQUE :

Numero de sueldos =  1
SUELDO ( 1 )=$  12478


SUMA TOTAL    $=  12478
```

$	CANT.
5000	2
1000	2
500	0
100	4
50	1
10	2
5	1
1	3

```
10 PRINT CHR$(4)
20 D=0
30 DIM A(4),B(4),X(4)
40 DIM Y(4),Z(4),W(4)
50 DIM L%(100),C%(100)
60 FOR N=1 TO 4
70 A(N)=0: B(N)=0
80 NEXT N
90 DEF FNAT$(L%,C%)=CHR$(155)+CHR$(C%+128)+CHR$(L%+128)
100 PRINT FNAT$(3,5);"A :      REVISTA  MICROBYTE""
110 PRINT FNAT$(5,5);"DE :      ROBERTO  CISTERNAS  ALISTE"
120 PRINT FNAT$(10,5);"D E S G L O S E     D E     S U E L D O S"
130 FOR E=1 TO 2000
140 NEXT E
150 PRINT CHR$(4)
160 PRINT"INDIQUE :":PRINT
170 INPUT "Numero de sueldos = ",M
180 FOR L=1 TO M
190 PRINT FNAT$(6,5);"SUELDO (";L;")=$    ?":INPUT "",S
200 PRINT FNAT$(7,0);"
210 PRINT FNAT$(6,18);S
220 PRINT FNAT$(19,35);"DATO CORRECTO     <S/N>"
230 Q$=INKEY$
240 IF Q$<>"S" AND Q$<>"N" THEN 230
250 PRINT FNAT$(19,35);"
260 IF Q$="N" THEN 190
270 D=D+S
280 K=5000: I=1000
290 FOR N=1 TO 4
300 X(N)=INT (S/K)
310 Y(N)=S-X(N)*K
320 Z(N)=INT (Y(N)/I)
330 W(N)=Y(N)-Z(N)*I
340 I=I/10: K=K/10
350 A(N)=X(N)+A(N)
360 B(N)=Z(N)+B(N)
370 S=W(N)
380 NEXT N
390 NEXT L
400 PRINT CHR$(4)
410 PRINT FNAT$(3,5);"SUMA TOTAL   $= ";D
420 PRINT FNAT$(8,5);"       $                         CANT."
430 PRINT FNAT$(9,5);"-------------------------------------------"
440 PRINT FNAT$(11,9);"5000"
450 PRINT FNAT$(12,9);"1000"
460 PRINT FNAT$(13,10);"500"
470 PRINT FNAT$(14,10);"100"
480 PRINT FNAT$(15,11);"50"
490 PRINT FNAT$(16,11);"10"
500 PRINT FNAT$(17,12);"5"
510 PRINT FNAT$(18,12);"1"
520 FOR N=1 TO 4
530 PRINT FNAT$(2*N+9,35);A(N)
540 PRINT FNAT$(2*N+10,35);B(N)
550 NEXT N
```

Renumerador para ZX-81

Omar Atanaa

Creo que a muchos usuarios como a mí, les ha sucedido que a veces, necesitan agregar una o más líneas a un programa BASIC y se encuentran con la dificultad de ponerlas entre dos líneas con numeros consecutivos, ejemplo 21 LET A = / 22 PRINT Una solución para esto sería cambiar el número de la línea 22 teniendo a veces que cambiar el numero de varias líneas hacia abajo, con la consiguiente pérdida de tiempo

Para superar este problema desarrolle un programa en ASSEMBLER Z80 o código de máquina, que cambia el número de todas las líneas, dejando un espacio de 10 entre estas, por ejemplo

	–queda–
2 LET A = 20	10 LET A = 20
15 LET B = INT (C)	20 LET B = INT (C)
16 PRINT A$	30 PRINT A$

De manera que ahora hay 10 lugares entre línea y línea, para agregar las que quiera, con sólo hacer funcionar esta rutina en código de máquina. Además esto le brinda al programa una apariencia más acabada, cuando está completo y quiere grabarlo

Otra cosa interesante es que esta rutina no ocupa lugar en el área del programa BASIC, ya que se almacena sobre el RAMTOP en forma automática

```
10 REM ZX-61
20 PDKE 16386, 226
30 PDKE 16389, 127
40 CLS
50 LET A$ =  '217D40010A007EFE76C87023
   71235E2356192379C60A4F76CE004718E9'
60 LET X = 32738
70 POKE X, 16• CODE A$ + CODE A$ (2) - 476
80 LET X - X + 1
90 LET A$ = A$ (3 TO)
100 IF A$ = "" THEN NEW
110 GOTO 70
120 SAVE "ZX-81"
130 RUN
```

Las líneas 20, 30 y 40 mueven la RAMTOP a la dirección 32738, –para 16K de memoria– luego las líneas 50 a la 110 cargan el programa en código de máquina, contenido en la variable A$, como un conjunto de numeros hexadecimales La línea 120 sirve para grabarlo y que una vez cargado se autoejecute

La explicación para el programa en ASSEMBLER Z80 es la siguiente

Cod. Hex	Instrucción	
217D40	LD HL, 16509	HL almacena la dirección donde comienza el programa BASIC
010A00	LD BC, 10	BC contiene el Nº que tendrá la 1ª línea del programa.
7E -Aqui-	LD A, (HL)	Esta parte determina que
FE76	CP 76	cuando se renumere todo el
C8	RET Z	programa se retome al BASIC
70	LD (HL), B	Aquí se le da el nuevo nú-
23	INC HL	mero de línea el programa
71	LD (HL), C	BASIC.
23	INC HL	
5E	LD E, (HL)	Acá se busca la dirección de
23	INC HL	la siguiente línea para cam-
56	LD D, (HL)	biarle el número
19	ADD HL, DE	
23	INC HL	
79	LD A, C	BC toma el nuevo valor de la
C60A	ADD A, 10	próxima línea, es decir, si
4F	LD C, A	antes era 10, ahora es 20 y
76	LD A, B	luego 30, así sucesivamen-
CE00	ADC A, 0	te
47	LD B, A	
18E9	JR AQUI	Vuelve a la rutina para cambiar el numero a la segunda línea

Una vez cargado el programa (con LOAD) la pantalla estará en blanco –recomiendo cargarlo cuando se conecte el computador– Como prueba, escriba un programa cualquiera en BASIC y realice RAND URS 32738 / ENTER. Así cada vez que necesite renumerar el programa, sólo tendrá que tipear RAND URS 32738 y ENTER

Usted puede cambiar el intervalo entre línea y línea, con la secuencia POKE 32759, N (donde N es un numero entero entre 1 y 255) / ENTER

Algo importante que no debe olvidar, es que una vez usado este programa en código de máquina, tiene que cambiar el numero de las instrucciones GOTO y GOSUB, ya que el programa no lo hace automáticamente

Serpiente de números

Si hay algo exasperante, es jugar un juego aparentemente simple y descubrir rápidamente que a pesar de su simplicidad somos incapaces de obtener un puntaje medianamente decente.

Ese es el caso del programa que nos ha enviado Juan J. Maten, de Santiago, quien se ha divertido exasperando al prójimo.

El estado del programa es breve y conciso (una cualidad apreciable entre los colaboradores) y su gracia principal es la velocidad con que va dejando fuera de combate a quienes nos animamos a jugar.

Tal como su nombre lo indica, en este juego el rol principal lo tiene una inocente víbora cuyo cuerpo está formado por números, los que se van generando en forma aleatoria. Naturalmente, el objetivo es hacer desaparecer al reptil antes que éste haga lo propio con nosotros.

Dos teclas ("f" e "i") son todo el arsenal con que se cuenta, por lo que hay que jugar con sapiencia haciendo avanzar un generador de números y cada vez que en este generador se encuentre un número que también lo contiene la serpiente, disparamos y lo eliminamos.

Eso fue lo que pensamos al ver el juego, sin embargo en la práctica resultó ser más difícil (y por tanto entretenido). Hay que tener cuidado con avanzar demasiado a prisa el generador de números pues podemos pasarnos del número que hay que eliminar en la víbora y el tiempo en este juego es oro. Por eso, no pierdan su tiempo y comiencen a tipearlo ya. No es seguro que ganen, pero se entretendrán.

```
  10 DIM d$(1,10): LET d$(1)=" "
  20 CLS : LET de=0: LET t$="1":
 LET lim=20: LET r=21: LET c$=""
: LET lmc=10: LET nmc=20: LET pu
nt=0: LET inc=0: LET ncg=0: LET
bono=0
  30 PRINT AT 6,10;"puntaje : ";
punt;" puntos";AT 11,10;t$;" . "
;d$(1, TO lmc-LEN c$-de);c$;"
"
  40 IF INKEY$="f" THEN GO SUB 1
000
  50 IF INKEY$="i" THEN GO SUB 1
500
  60 IF r>lim THEN LET r=0: GO S
UB 1600: IF LEN c$+de>lmc THEN G
O TO 100
  70 LET r=r+1: GO TO 30
 100 CLS : PRINT AT 11,1;"lo sie
nto, Ud. ha perdido";AT 12,1;"su
 puntaje fue de ";punt;" puntos"
;AT 13,1;"desea jugar otra vez ?
 (s/n)"
 120 IF INKEY$="s" THEN GO TO 20
 130 IF INKEY$="n" THEN CLS : ST
OP
 140 GO TO 110
1000 LET lc=LEN c$: LET r=r+lc
1010 FOR i=1 TO lc
1020 IF c$(i)=t$ THEN GO TO 1070
1030 NEXT i: RETURN
1070 LET c$=c$(1 TO i-1)+c$(i+1
TO lc)
1080 LET punt=punt+1+inc: PRINT
AT 11,10;t$;" . ";d$(1, TO lmc-L
EN c$-de);c$;" "
1090 IF t$="*" THEN LET punt=pun
t+2+inc: BEEP .2,7
1100 BEEP .1,11: RETURN
1500 IF t$="*" THEN LET t$="1":
LET r=r+2: RETURN
1510 IF t$="9" THEN LET t$="*":
LET r=r+2: RETURN
1520 LET t$=STR$ (VAL t$+1): LET
 r=r+2: RETURN
1600 LET r=r+2: LET ncg=ncg+1: I
F ncg>nmc AND c$="" THEN GO SUB
1650
1605 IF ncg>nmc THEN LET de=de+1
: RETURN
1610 LET k$=STR$ (INT (RND*10)):
 IF k$="0" THEN LET k$="*"
1620 LET c$=c$+k$: RETURN
1650 CLS : LET bono=bono+10: LET
 punt=punt+bono
1660 PRINT AT 11,5;"muy bien, ha
s pasado";AT 12,5;"a otra serie.
..";AT 14,10;"BONO ";bono
1670 FOR n=1 TO 9: BEEP .2,20-n:
 BEEP .2,n+11: NEXT n: PAUSE 60
1680 CLS : LET de=0: LET c$=""
LET lim=lim-3: LET lmc=lmc-1: LE
T ncg=0: LET nmc=nmc-5: LET inc=
inc+1
1690 IF lim<=10 THEN LET lim=20
1695 IF nmc<=5 THEN LET nmc=15
1700 IF lmc<=3 THEN LET lmc=6
1710 RETURN
```



Orden Alfabético

Entre las consultas que a diario llegan a nuestra revista, en varias oportunidades se nos ha pedido expliquemos con ejemplos la forma en que el Atari maneja variables alfanuméricas

Esta vez, aprovecharemos el programa enviado por Arnulfo Oelker, de Concepción, para finalmente hacerlo El programa consiste en aceptar una lista de palabras, las que el computador finalmente entregará ordenadas en orden alfabético Para usar el programa, ingrese todas las palabras que desea y para terminar escriba FIN El computador entregará la lista de palabras ordenadas en pantallas de 15 Para ver la siguiente pantalla es necesario presionar SELECT

En la línea 10 del programa, se ha dimensionado un arreglo alfanumérico, A$, con espacio para 10 000 caracteres En éste se irá guardando la lista de palabras en una sola cadena Las palabras son originalmente ingresadas en la variable C$ (línea 120) y de inmediato se invoca la subrutina en la línea 1000, cuyo objetivo es lograr que todas tengan el mismo largo Esto se obtiene agregando espacios hasta que la palabra tenga un largo determinado

La instrucción que hace esto es la 1010, la que en palabras quiere decir 'en la variable B$, en la posición largo de B$ más uno hay que poner un P$, el que está definido en la línea 20 como un espacio vacío. De este modo se logra la concatenación de strings en el Atari Por poner otro ejemplo, si A$ = "MANI" y B$ = "COMIO", entonces la instrucción A$(LEN(A$) + 1) = B$ nos permitirá juntar en A$ las dos variables quedando A$ = "MANICOMIO"

En el caso de este programa, el orden alfabético se realiza mediante punteros que van almacenados en una matriz, lo que dificulta un tanto la explicación, pero digamos que para ordenar, es necesario comparar cada palabra con el resto y para eso debemos separar la lista larga en las palabras individuales

Como todas las palabras en este caso tienen el mismo largo (30), hay que dividir A$ en segmentos de 30 caracteres Para esto se utiliza una variable auxiliar, en este caso alguna de las variables dimensionadas en 30 en la línea 10

En las líneas 300-400, el programa divide la cadena A$ en segmentos y los compara mediante la instrucción A$(X, X1) en que X representa al carácter inicial y X1 el final de cada palabra Como cada palabra ocupa 30 caracteres, en cada vuelta del ciclo FOR-NEXT los valores de estas variables son incrementados en 30

Para resumir, a pesar de que el Atari no maneja arreglos de variables alfanuméricas, no es difícil simular el uso de éstas mediante un string largo, que incluye a todos los elementos del arreglo, los que son concatenados y separados a voluntad

```
0 CLR
10 DIM A$(10000),B$(30),C$(30),D$(30),E$(30),R$(5),P$(5)
20 B=10:G=38:R$="    ":W1=1:W=4:P$=" "
30 A$="                  "
40 K1=1:K=1:X2=10:X3=38
90 GRAPHICS 0
100 POSITION 10,2:? " ORDEN ALFABETICO "
110 POSITION 2,4:? "INGRESE PALABRAS A ORDENAR:"
120 POSITION 2,7:? "PALABRA ";K1;".- ";:INPUT C$:IF C$=
    "FIN" THEN 230
130 C=LEN(C$):GOSUB 1000
140 C$(LEN(C$)+1)=B$
150 A$(B,G)=C$:B$=""
160 B=B+30:G=G+30:K1=K1+1
170 POSITION 1,7:? "
"
180 A$(LEN(A$)+1)=" "
200 GOTO 120
230 POKE 752,1:POSITION 9,17:? "UN MOMENTITO..."
240 A$(LEN(A$)+1)="                                        "
250 X1=8:B=B-30
290 Z=65:O$=CHR$(Z)
```

```
300 FOR X=10 TO B STEP 30
310 X1=X1+30
320 E$=A$(X,X1)
330 IF ASC(E$)=Z THEN A$(X2,X3)=E$:X2=X2+30:X3=X3+30
400 NEXT X
410 X1=8:Z=Z+1:k2=K2+1
420 IF Z>90 THEN 500
430 GOTO 300
500 X1=8
510 GRAPHICS 0:POKE 752,1
520 POSITION 3,1:? "LISTADO ALFABETICAMENTE ORDENADO #";W1
530 POSITION 3,2:? ""
550 FOR X=10 TO 8 STEP 30
560 X1=X1+30
570 E$=A$(X,X1)
580 D$=A$(X+30,X1+30)
590 IF ASC(E$)=ASC(D$) THEN 700
600 POSITION 2,W:? K;".-";R$;A$(X,X1):K=K+1:W=W+1
610 IF K>9 THEN R$="  "
620 IF K>99 THEN R$=" "
630 IF K>999 THEN R$=""
640 IF W=18 THEN 2000
650 NEXT X
670 OOTO 900
700 FOR I=1 TO 28
710 E=ASC(E$(I,I))
720 D=ASC(D$(I,I))
730 IF E<D THEN 600
740 IF E>D THEN A$(X,X1)=D$:A$(X+30,X1+30)=E$:OOTO 600
750 NEXT I
760 GOTO 600
900 POSITION 5,19:? "PRESIONE <START> PARA COMENZAR"
910 A=PEEK(53279)
920 IF A=6 THEN RUN
950 GOTO 910
1000 FOR X=1 TO 28-C
1010 B$(LEN(8$)+1)=P$
1020 NEXT X
1030 RETURN
2000 A=PEEK(53279)
2010 IF A=5 THEN 2050
2020 POSITION 4,19:? "PRESIONE <SELECT> PARA CONTINUAR"
2030 GOTO 2000
2050 W1=W1+1:GRAPHICS 0:POKE 752,1
2060 POSITION 3,1:? "LISTADO ALFABETICAMENTE ORDENADO #";W1
2070 POSITION 3,2:? "":W=4
2100 NEXT X
2120 GOTO 900
```

Autonumerador para el VIC-20

Andrés E. Arancibia

Este utilitario permite definir una secuencia de autonumeración en el VIC-20 Basta cargarlo desde el cassette, ingresar SYS 840, luego el numero de línea inicial, despues el incremento y comienza la secuencia

El Basic del VIC-20 no incluye un numerador automático, cuya utilidad es considerable cuando se desea ingresar un programa largo La gran gracia de este utilitario en lenguaje de máquina es que no disminuye la ya escasa memoria RAM del VIC (3,5 K), ya que se carga en el área del buffer del cassette Permaneciendo allí hasta que use algun comando relativo al cassette (Load Save, o Verify)

El primer paso es ingresar el programa, teniendo especial cuidado en las instrucciones DATA. Luego se ejecuta el programa, si no hay errores en las DATA aparecerán dos líneas en la pantalla

```
NEW
A$ =      FOR I = 833 TO 1010
A$ = A$ + CHR$ (PEEK (I) ) NEXT I
SAVE A$
```

Es conveniente SALVAR el programa en este paso A continuación es necesario ejecutar las líneas que aparecieron en el paso anterior Para esto basta volver a dichas lineas con el cursor y presionar RETURN una vez encima de ellas (NEW primero, luego la otra), al ser solicitado presionar RECORD y PLAY en la cassetera. No hay que preocuparse por los gráficos que aparecerán durante el SAVE

Ahora, hay que probar el programa, cargando el programa recién grabado, mediante un Load

Luego de la carga, el programa se inicie con un SYS 840 A continuación aparecerá el cursor en la línea siguiente El programa está esperando el número de la línea inicial Al ingresarlo, el cursor pasará a la siguiente linea y esperará el incremento, una vez ingresado este ultimo aparecerá la secuencia inicial en la línea siguiente De ahi en adelante se puede ingresar el programa Cada vez que se presione RETURN, que indica el final de la linea, aparecerá a continuación el numero de la línea siguiente.

```
10 FOR I=833 TO 1010
20 READ A
30 T = T + A
40 POKE I, A
50 NEXT I
60 IF T <> 22841 THEN  PRINT"ERROR EN
DATA" STOP
70 PRINT"NEW"
80 PRINT"A$="CHR$(34)CHR$(34)":FORI=
833TO1010 A$=A$+CHR$(PEEK(I)):NEXT
I SAVE A$
"
90 END
1000 DATA13,83,217,57,52,50,0,32
1010 DATA199,3,134,252,133,251,169,13
1020 DATA32,210,255,32,199,3,134,254
1030 DATA133,253,169,13,32,210,255,166
1040 DATA251,165,252,32,205,221,32,207
1050 DATA255,201,92,240,63,201,13,208
1060 DATA245,165,251,24,101,253,133,251
1070 DATA165,252,101,254,133,252,162,4
1080 DATA169,145,157,119,2,202,208,250
1090 DATA169,13,141,123,2,141,124,2
1100 DATA162,6,134,198,173,134,2,133
1110 DATA1,173,15,144,74,74,74,74
1120 DATA41,7,141,134,2,169,65,160
1130 DATA3,32,30,203,96,165,1,141
1140 DATA134,2,169,13,32,210,255,32
1150 DATA210,255,162,66,169,20,32,210
1160 DATA255,202,208,250,240,153,169,0
1170 DATA160,0,32,162,219,32,207,255
1180 DATA201,13,240,22,41,207,201,10
1190 DATA16,11,72,32,226,218,104,32
1200 DATA126,221,56,176,232,162,22,76
1210 DATA58,196,32,155,220,166,100,165
1220 DATA101,96
```

Para terminar la autosecuenciación basta que aparezca dentro de la línea el carácter £ (libra esterlina) Dicha linea NO será incluida en el programa

Si se desea retomar la secuencia (Línea donde apareció £) se puede dar un SYS 864 o bien recomenzar con un SYS 840

No es recomendable editar lí-

neas mientras se está en la autosecuencia.

Notas de operación

Si se ingresa un valor no numerico para el valor inicial y/o el incremento se producirá un "TYPE MISMATCH ERROR"

Cabe recordar que el VIC no acepta numeros de línea mayores que 63999, si se ingresara un número mayor será un error de sintaxis En estas ocasiones, así como a veces cuando se editan líneas anteriores dentro de la autonumeración, ésta se interrumpe y el cursor desaparece (en realidad queda con el color de la pantalla) Para recobrarlo basta presionar el CTRL-COLOR que se estaba usando

Es posible cargar el programa en el buffer del cassette por medio de un SYS 63407 La ventaja de este método es que no borra el programa que está en memoria

Como el carácter £ es usado como escape no es posible ingresar una línea que lo necesite durante la autonumeración Hay dos soluciones Se puede ingresar la línea que lo contenga una vez que se termine de ingresar el programa. La otra sería cambiar el carácter de escape, para esto bastaría con modificar la DATA 1050 En esa línea el tercer numero es 92 (que corresponde al carácter £, ver manual del computador Apéndice J pág. 146) Al cambiarlo por otro número, por ejemplo el 95 (flecha hacia la izquierda) éste será el nuevo carácter de escape. Obviamente, al cambiarlo aparecerá el mensaje ERROR EN DATA, ya que la suma cambia Para solucionar esto basta cambiar la línea 60 agregando la diferencia (95-92 = 3) a 22842 quedando en 22845

Este programa sirve también para eliminar lineas (borrarlas) Basta con dar un RETURN a continuación de la secuencia y la línea será borrada Asimismo si aparece un número que no se desea ingresar (por ejemplo, cuando se copia un listado que aparece en una revista como el editor de assembler, Microbyte Nº 7) basta dar un RETURN y aparece la secuencia siguiente

Nuevos trucos para el VIC 20

José Luis López Castillo

Les presento algunos trucos nuevos para que puedan ser usados en programación y en juegos para el VIC 20 La mayoría de estos "trucos" consisten en utilizar ciertas localizaciones de memoria que normalmente no son conocidas o bien sus funciones no están claras

TRUCO 1: Cuando se trabaja con caracteres modificados, especialmente en los juegos no conviene que el set de caracteres pase de mayúsculas a minusculas o viceversa al apretar las teclas Commodore y Shift

Para deshabilitar esta función tipee

POKE 657, 128

Para volver a habilitar el otro set se debe colocar el valor cero en la localización de memoria antes indicada

TRUCO 2: Una de las funciones menos utilizada por los VIC-iosos es la instrucción WAIT Esta instrucción detecta la variación de una cierta localización de memoria realizando una operación "OR" con el argumento Si existe un segundo argumento, el resultado anterior se realiza una operación "AND" Si el resultado es cero volverá a consultar, y si no el programa continua Esto hace que sea una manera muy económica (en bytes) para parar un programa, en vez del muy socorrido

100 GET A$ IF A$ = ""THEN 100

Ya que utilizando los WAITs esta misma línea puede quedar de la forma

100 WAIT 653, A

Si A = 1 el programa continuará si es presionada la tecla Shift si A = 2 la tecla Commodore será la que continue el programa y si A = 4 la tecla CTRL continuará el programa

Otra localización utilizable para estos fines es la 198.

100 POKE 198, 0 : WAIT•198, 1 : POKE 198, 0

Para lo cual basta apretar cualquier tecla

TRUCO 3: Lo he llamado "Instrucciones Automáticas", ya que al colocar ciertos valores en el buffer de teclado podemos colocar cualquier instrucción El buffer posee 10 bytes (localizaciones 631-640) y trabaja con un puntero que indice el número de caracteres en la cola

631	632	633	634		640
R	U	N	(RETURN)	.....	

(198) Nº CARACTERES EN LA COLA

En la cola debemos colocar los valores ASCII de las letras o instrucciones que deseemos que aparezcan en forma "automática" cuando el VIC deje de trabajar

Para el ejemplo de la figura tipee el siguiente listado

LISTADO 1 Inst Automáticas

```
10 FOR I = 631 to 634 READ J POKE I,J NEXT
POKE 198, 4
```

```
20 DATA 82, 85, 78  13
30 LIST
```

Recuerde escribir como ultimo carácter el correspondiente a haber apretado la tecla RETURN (valor ASCII 13)

TRUCO 4: Lineas Invisibles

En mi artículo anterior Trucos para el VIC 20", vimos la manera de no listar un programa Pero, ¿si sólo queremos hacer invisibles algunas líneas y no todo el programa?

La solución a esta inquietud está en el mágico REM Por ejemplo despues de la línea 20, detrás del numero 13 en el listado 1 tipee los dos puntos, escriba REM y abra dos comillas, coloque el cursor sobre la ultima comilla e inserte aproximadamente 15 espacios Ahora apriete la tecla DEL Si lo ha hecho correctamente tendrá 15 letras T en reversa Grabe la línea y listee   ¿Dónde se fue la línea?, pese a no verla la línea existe y el listado sigue funcionando perfectamente Pero ni Ud ni nadie la podrá ver más .

TRUCO 5: No debería ser considerado como tal, pero nació como un desafío puesto (tácitamente) en la MICROBYTE de octubre El listado 2 presenta un trazador, como el publicado para el C-64 Pero éste indica el número de la línea por cada instrucción en dicha línea en el formato siguiente

Para 10 X = 1 mostrará <10>

Las funciones de este tracer son las mismas que las indicadas para el tracer del C-64

Listado 2 TRACER VIC-20

```
100 PRINT CHR$ (147)  TRAZADOR VIC-20'
110 D = 828   REM DIRECCION INICIAL
120 FOR I = D TO D + 53  READ J  POKE I J  NEXT
130 PRINT  SYS  , D + 22,  ACTIVA TRACER'
140 PRINT 'SYS', D + 43   DESACTIVA TRACER
150 DATA 165  58, 201  255  240, 13, 169, 60
160 DATA 32, 210, 255, 32, 201  221, 169, 62
170 DATA 32, 210  255, 108, 251, 0, 173, 8, 3
180 DATA 133, 251, 173, 9, 3, 133, 252
190 DATA 169, 60  141, 8  3  169  3  141  9  3
200 DATA 96  165  251, 141  8  3  165  252, 141  9, 3  96
```

TRUCO 6· List Controlado

Este truco consiste en interceptar el vector LIST para que cuando se escriba la instrucción LIST sólo liste cuando se presione la tecla L y sólo mientras esté presionada El vector se encuentra en las localizaciones 774 y 775 (HEX $ 306 y $ 307), y apunta a la localizacion 50970 (HEX $C43A) Cambiando el vector LIST podemos colocar cualquier rutina en lenguaje de máquina, guardando los valores del acumulador y del registro-Y El listado Assembler (listado 3) es un ejemplo de cómo podemos interceptar al vector LIST

El LIST controlado es presentado en BASIC en el listado 4

Listado 3 Subrutina Assembler

```
PHA
TYA
PHA
K LDA $C5
CPX #$15
BNE K
LDA #$00
STA $C6
PLA
TAY
PLA
JMP $C43A
```

Listado 4 LIST Controlado

```
5 POKE 774  60  POKE 775  3  REM VECTOR LIST
APUNTA LOC  828
10 FOR I = 828 TO 846  READ J  POKE I, J  NEXT
20 DATA 72  152, 72  164, 197, 192  21  208  250,
169, 0, 133, 198  104, 168, 104, 76  26  199
30 NEW
```

El listado 4 se borra automáticamente al correrlo, por lo que se debe grabar previamente Para deshabilitar el vector que apunta a nuestra rutina tipee

POKE 774, 26   POKE 775, 199

Una de las localizaciones importantes es la 197 y 203 que designan qué tecla ha sido presionada, y no qué carácter, especialmente util en juegos (ambas localizaciones tienen la misma función) Tendrán el valor 64 si ninguna es presionada y por ejemplo el valor 21 (hex $15) si la letra L es presionada (usada en el listado 3)

Quisiera expresar mi agradecimiento al Sr E Ahumada M por sus listados del ensamblador y desensamblador para el VIC 20, publicados en numeros anteriores de MICROBYTE, ya que me han sido de muchísima utilidad, en especial en la confección de los listados 2 y 4

Pasando a otro tema ¿Algun VIC-ioso sabe usar la función USR? yo aun no sé y me tiene bastante "picado" Si alguien tiene la solucion puede darla por medio de la revista para que nos sirva a todos, de antemano gracias

RESULTADOS CONCURSO RELAX

[illegible]

Ejecución de Programas en MS-DOS

Luciano E. Chiang
Ingeniero Civil Mecánico

Programas

Al hacerse cargo el sistema operativo MS-DOS del microcomputador correspondiente, el programa procesador de comandos Command Com (ver artículo de noviembre) queda residente a la espera de que el usuario ingrese un comando o pida la ejecución de un programa a través del teclado

Cabe hacer mencion que los comandos del sistema operativo se ejecutan por intermedio de rutinas que permanecen residentes en memoria principal (ROM o RAM) o por programas (utilitanos) que por tamano y frecuencia de uso no es conveniente que permanezcan residentes Estos ultimos estan a disposición del usuario en memoria secundaria (disco o diskette) y son invocados a discreción de éste

Si la instrucción que ingresa el usuario desde el teclado no correspondiera a un comando cuya rutina reside en memoria principal el sistema operativo deduce que se trata de la invocación de un programa que se desea ejecutar para lo cual es necesario traerlo desde disco o diskette

El nombre de un archivo en MS-DOS PC-DOS, en particular el de un archivo ejecutable o programa, consta básicamente de dos partes En primer lugar se tiene el nombre propio del archivo el cual, puede tener un largo máximo de 8 caracteres En seguida se tiene la extensión, la que puede tener un largo máximo de hasta tres caracteres y que sirve principalmente como argumento mnemónico de las características del archivo El nombre completo del archivó consiste en la yuxtaposicion de ambos nombres separados por un punto ( ) para diferenciarlos entre si

Entre los archivos ejecutables, MS-DOS reconoce tres tipos y los identifica por la extension que llevan, a saber

- BAT programas de tipo batch (grupo de comandos)
- COM programas de tamano fijo tipo comando
- EXE programas de tamano variable

Prioridad en la Ejecución

MS-DOS presume a cualquier archivo con alguna de las tres terminaciones antenores como un programa ejecutable Dos programas distintos podrán tener el mismo nombre propio pero deberán diferenciarse en la extensión la que indicará tanto el tipo de programa como el orden de prioridad para su ejecución La primera prioridad la tienen los programas con terminación BAT Enseguida tienen prioridad los programas con extensión COM y la ultima prioridad la tienen los programas con terminación EXE

Sin perjuicio de la prioridad de los programas, es posible pedir la ejecucion de cualquiera, obviando el orden establecido, si se utilizan las funciones apropiadas de MS-DOS Estas sin embargo no estan directamente disponibles al usuario, y su uso es materia de estudio más avanzado

Bloque de Datos

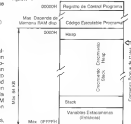
En la mayoría de los casos los programas contendrán variables que manejar Estas podran ser permanentes o transitonas Las variables permanentes o estacionarias son aquellas que existen durante todo el tiempo que se este ejecutando el programa las transitorias, son las que podrán existir por un determinado periodo de tiempo lapso que es controlado por el programa mismo

Todas las variables al momento de ejecutarse un programa están agrupadas en un bloque de datos que se carga en memoria en forma contigua al código ejecutable El tamaño del bloque de datos se define por el usuario en el caso de programa en assembler o por el compilador en el caso de lenguajes de alto nivel Sin embargo por razones constructivas del microprocesador, se limita el tamaño a un máximo de 64 KB

Las variables permanentes son ubicadas en la porción superior del bloque de datos (ver Fig N ° 1) Lo que resta disponible se utiliza para configurar las estructuras de pila de Stack y Heap La primera lleva un control de las subrutinas y funciones activas en un determinado instante y simultáneamente de sus variables internas La segunda lleva el control de las variables de puntero que se hayan creado (ver Programas con terminación EXE)

Diagrama del Bloque de Datos de un Programa

Figura N ° 1



Programas con Extensión BAT

Estos programas son archivos de texto almacenados en código ASCII que contienen una secuencia de comandos para ser ejecutados. Esta extensión proviene del concepto inglés de "Batch" que implica "grupo de comandos". El sistema operativo lee el contenido de estos archivos y lo ejecuta secuencialmente. La utilidad que prestan este tipo de programas es obvia. Permite realizar funciones de administración de archivos y periféricos que estén encadenadas y que sean de uso repetitivo. Del mismo modo permiten la ejecución enlazada de programas sin la intervención del usuario.

No es del caso entrar en más detalle al respecto de estos archivos pues amplia documentación se puede encontrar en los manuales de MS-DOS y literatura afín.

Programas con Extensión COM

Este tipo de programas se caracterizan por el hecho de que no utilizan las estructuras de pila de Stack y Heap. Como consecuencia, durante su ejecución el tamaño se mantiene constante. La ventaja de esta modalidad radica en el hecho de que el espacio de memoria requerido es menor a su contraparte (programa EXE), pues no se requiere agregar al código ejecutable las rutinas necesarias para administrar las mencionadas estructuras, ni reservarles el bloque de memoria correspondiente a su uso. Generalmente estos programas son de un pequeño tamaño, realizan funciones específicas de comandos y no manejan gran cantidad de variables.

La obtención de programas COM se logra en primer lugar asegurando que la dinámica interna de éste no requiera el uso de las estructuras de pila de Stack y Heap. Ello prescribe normalmente los lenguajes de alto nivel como lo son Pascal, Fortran o C, lo que para efectos prácticos nos deja limitados al lenguaje assembler.

Luego de ensamblar el programa fuente (texto del programa) se efectúa el enlace (linking) de éste. Como resultado se obtiene un archivo ejecutable con estructura EXE. Para transformarlo a estructura COM es necesario correr el programa EXE2BIN.EXE entregándole como parámetro el nombre del programa a convertir.

Programas con Extensión EXE

Los programas que poseen esta estructura corresponden a los archivos ejecutables de uso más frecuente. Tienen como característica principal el uso de las estructuras de pila de Stack y Heap.

En programas que manejen una cantidad grande de datos, el uso de las estructuras mencionadas es muy provechoso. Permiten en primer lugar, la creación de variables dinámicas (variables de puntero) durante la ejecución del programa. Así, la administración de la memoria destinada a alojar variables se hace más eficiente, pues no se requiere reservar espacio para una variable cuyo uso no es del todo seguro, a menos que el programa en forma interna lo determine necesario. A medida que se van creando las variables dinámicas, el espacio destinado a ellas se va reservando en forma ascendente desde las posiciones más bajas de memoria en el bloque de datos.

Por otro lado, el Stack permite la existencia temporal de variables. En efecto, las variables internas de las rutinas, procedimientos o funciones en lenguajes de alto nivel como Pascal y C, sólo existen mientras la rutina esté activa, y sólo podemos referenciarlas desde el interior de tales rutinas (a menos que explícitamente se indique lo contrario).

Una variable puede existir temporalmente pues al ser activada la rutina que la contiene, el programa crea un registro de control de ésta y lo apila en forma descendente en la estructura de pila de Stack (donde lo indique el puntero del Stack). Este registro de control contiene entre otras cosas un puntero que indica hacia la rutina precedente o llamadora y espacio de reserva para las variables internas.

El lector sagaz se preguntará si, existiendo un bloque de datos de tamaño finito, y creciendo desde sus extremos en sentidos opuestos el Stack y el Heap, ¿cómo se resuelve el conflicto si ambas pilas se encuentran y traslapan?

Bástele saber a ese lector que ello es poco probable que ocurra. El bloque de datos, que contiene al Heap y al Stack, se crea con el tamaño suficiente para permitir el crecimiento adecuado de ambas pilas. Sin embargo, en los casos en que se presentara el problema, el programa invariablemente abortará indicándose un mensaje al efecto en la pantalla.

Direccionamiento de Memoria

Tanto el microprocesador Intel 8088 como el 8086, típicos de los sistemas MS-DOS, pueden accesar en forma directa 1 MB de memoria a través de sus 20 pins del bus de direcciones. Sin embargo, dado que son microprocesadores con registros internos de 16 bits (los que permiten un acceso máximo de 64 KB) fue necesario elaborar un mecanismo para hacer posible el acceso al total de la memoria disponible.

La solución escogida fue la de utilizar dos registros (ver Fig. 2) para una determinada posición de memoria. El valor del primer registro corresponde a la dirección de segmento o de los 16 bits más significativos de los 20 que se requieren en el bus de direcciones. El valor del segundo registro corresponde a una dirección de desfase (offset) y define a los 16 bits menos significativos de la dirección absoluta.

Nótese que para un determinado segmento es posible accesar hasta 64 KB de memoria al hacer variar en todo su rango al registro de desfase correspondiente. Por otro lado es de interés señalar que un segmento por sí solo define un bloque de 16 bytes de memoria.

Existen varios registros de segmento y de desfase con el objeto de otorgar mayores facilidades en la administración de la memoria. Para los segmentos se utilizan cuatro y se usan de la siguiente manera:

- CS: registro de segmento para el código ejecutable
- DS: registro de segmento para el bloque de datos
- SS: registro de segmento de inicio de Stack
- ES: registro de segmento auxiliar de datos

Continúa en pág. 38



El Nuevo MAI Basic Four 2000. La Síntesis Perfecta de la Revolución de Los Super Microcomputadores y La Confiabilidad de lo Probado.

El Sistema de Administración MAI BASIC FOUR 2000® combina la potencia de un supermicrocomputador multiusuario con la disponibilidad de software comercial y profesional de alta calidad probado en cientos de instalaciones en Chile y miles en todo el mundo

En el Sistema 2000 converge la tecnología más reciente y la compatibilidad con toda la línea de computadores MAI BASIC FOUR.

Le hemos dado a nuestro Sistema Operativo tipo UNIX™ un carácter amistoso para que sea confiable a personas que no tienen ninguna experiencia en computación Este Sistema se llama BOSS/IX

Características Sobresalientes

- Procesador Central Ultra compacto con 1 MB de memoria
- Capacidad en discos magnéticos desde 22 MB hastra 240 MB
- CPU de alta velocidad motorola 68010
- Cinta Magnética Streamer en Cartridge de 43 MB y alta velocidad de respaldo
- Hasta 14 terminales locales o remotos (más de 600 en Red local)
- Business BASIC Nivel IX
- Sistema Operativo BOSS/IX™
- Transportadores de aplicaciones y archivos desde los niveles Basic Four anteriores (S/10, 110–730, MAI 8000, BBI y BBII)
- Sistema Generador de Aplicaciones ORIGIN™ de cuarta generacion
- Sistema de Bases de Datos Relacionales INFORMIX™
- Red Local MAGNET™ Hasta 63 Sistemas MAI 2000 interconectados

LOGICA

Oficina Principal Vecinal 61 Teléfonos 2312626-2314627 / Servicio Técnico Eleodoro Yáñez 1212 Teléfono 749094 / Sistemas y soporte Lyon 1735 Teléfono 44150

Los registros utilizados para el desfase de la dirección son los siguientes

- IP registro de desfase de la instrucción a ejecutar (puntero de instrucción)
- SP registro de desfase del Stack (puntero del Stack)
- DI registro de desfase de índice de destino (para el intercambio de datos en memoria)
- SI registro de desfase de índice de fuente (para el intercambio de datos en memoria)
- BP registro de desfase de base

Otros registros que se usan tanto como registros de desfase como para almacenar datos son DX y BX

Direccionamiento en el 8088/8086



Por ejemplo
si
desfase 1234H
segmento - 5678H
entonces

01234H
+ 56780H
579B4H Dirección Absoluta

Fig 2a

Registros en el 8088/8086

Registros de Datos: AH, AL (AX Acumulador); BH, BL (BX Base); CH, CL (CX Contador); DH, DL (DX Datos)

Registros de Desfase: SP Puntero de Stack; BP Puntero de Base; SI Puntero de Índice Fuente; DI Puntero de Índice Destino; IP Puntero de Instrucción

Registros de Segmento: CS Segmento de Código; DS Segmento de Datos; SS Segmento de Stack; ES Segmento Auxiliar

Señales de Estado

Fig 2b

Mecanismo de Carga de un Programa

Cuando MS-DOS carga un programa en memoria principal para su ejecución, este procede a efectuar una serie de tareas administrativas antes de entregar el control al programa mismo

La primera acción que toma es reservar toda la memoria disponible para la ejecución del programa sin perjuicio de que realmente se requiera o no. Ello se realiza restando del total de la memoria RAM disponible el espacio utilizado por el sistema operativo y otros utilitarios residentes en ese instante. Del total de la memoria asignada al programa se destinan 256 bytes (100H) para el registro de control del programa (ver Fig 3). Este registro de control es utilizado entre otras cosas para

- indicar vector a la dirección de término normal del programa
- indicar vector a la dirección de término de ejecución por error fatal: división por cero, intento de acceso a archivo inexistente, etc
- indicar vector a la dirección de término de ejecución cuando se oprima Ctrl-Break para abortar el programa (interrupción por el usuario)
- indicar valor de segmento más alto siguiente al espacio ocupado por programa
- bloque de control de archivos (se utiliza para el control de los archivos abiertos por el programa)
- Buffer de transferencia de datos desde diskette disco

Registro de Control de Programa

Hex

00: Int 20H | Dirección de Segmento a continuación de Programa | Número de Bytes en el segmento

08: Terminación Normal de Programa IP CS | Ex. por Ctrl Break IP

10: Ex. por Ctrl Break CS | Finalización por Error Fatal IP CS

2C: Dirección de Segmento de Parámetros DOS

5C: Bloque de Control de Archivos (36 bytes)

80: Buffer de Transferencia desde disco diskette (128 bytes)

Fig 3

Luego de construir el registro de control de programa (Program Segment Prefix), el sistema operativo procede a cargar las distintas partes del programa a las posiciones de memoria que corresponda. Comienza en primer lugar por el código ejecutable para posteriormente seguir con el bloque de datos

Cuando el programa tiene estructura COM el bloque de datos tiene tamaño fijo pues todas las variables son permanentes. En el caso de programas con estructura EXE, el SO (sistema operativo) carga las variables que son permanentes en las direcciones más altas del bloque de datos. De la memoria disponible restante se procede a inicializar al puntero del Stack a la dirección más alta y el Heap a la dirección más baja dentro del bloque de datos

Finalmente el SO procede a inicializar los registros CS e IP con la dirección de segmento y desfase del inicio del programa y los registros DS y SS al valor de comienzo del bloque de datos. Enseguida el procesador salta a la instrucción contenida en la dirección CS: IP y recién el control del sistema pasa al programa a ejecutar

Ejecución de Programas

Cuando el control del microcomputador pasa al programa del usuario, el SO permanece residente (latente) pero inactivo. Así, cuando el programa requiere del intercambio con alguno de los periféricos, v.b. disco, diskette, pantalla, teclado, etc., el control vuelve momentáneamente a MS-DOS

El intercambio de control se realiza a través de interrupciones. Esta modalidad otorga gran flexibi-

lidad para el programador así como para los fabricantes de hardware y software no estándar u opcional

La forma en que estas interrupciones funcionan consiste en el llamado de subrutinas en forma indirecta Cada interrupción posee un numero que corresponde a la posición de memoria (vector) donde se almacena la dirección de la rutina a ejecutar Así por ejemplo, la interrupción 20H obliga al procesador 8088/8086 a buscar en la tabla de interrupciones la entrada numero 20H (en posición 0000 0080H) y desde allí obtener la dirección en memoria de la rutina a ejecutar

Al comienzo de la ejecución de la rutina, el procesador apila en el Stack la dirección de retorno al programa del usuario, de tal manera que al término de ésta se pueda reasumir la ejecución del programa a continuación del comando que llamó a la interrupción

Terminación de la Ejecución

Al finalizar el programa usuario, es preciso devolver el control del procesador al sistema operativo Con tal objeto el usuario, en el caso de programa assembler o compilador en el caso de lenguaje de alto nivel deberá tomar las precauciones para que ello ocurra en forma correcta

Es posible terminar un programa de varias maneras La más comun es utilizando la interrupción 20H En este caso el control vuelve al sistema operativo y el espacio de memoria utilizado por el programa es liberado Los archivos aun abiertos son cerrados y la parte transiente del procesador de comandos se vuelve a cargar

La otra forma común para finalizar los programas es a través de la interrupción 27H Bajo esta modalidad, el programa usuario permanecerá residente en memoria y podrá ser llamado en cualquier instante sin tener que recurrir a almacenamiento secundario

PC-DOS en su versión 2 0 ha añadido otras dos formas de terminación las que se invocan a través de las funciones DOS incorporadas a la interrupción 21H En especial merece mención la función 4CH, la que entrega un código de finalización una vez terminado el programa En condiciones de terminación normal, este código tendrá un valor cero Cuando la terminación ha ocurrido debido a un error fatal, entonces este código asume un valor correspondiente a la causa del error Esta herramienta es muy importante cuando se requiere encadenar la ejecución de programas y los resultados generados por unos son utilizados posteriormente por otros

Conclusionea

Se ha visto en este artículo aspectos generales acerca de la mecánica de ejecución de programas en el sistema operativo MS-DOS/PC-DOS Reconociendo que por limitaciones de espacio no es posible entrar en detalles muy específicos, la intención ha sido entregar una visión bastante amplia para que en seguida el lector pueda recurrir a la literatura pertinente y ahondar en los detalles que le interesen más En todo caso en los próximos numeros nos permitiremos profundizar con más rigurosidad en temas de interés acerca de este popular e interesante sistema operativo

Bibliografía

1 The IBM PC-DOS HANDBOOK
Richard Allen King
Editorial Sybex, Berkeley, California
1983

2 The IBM Personal Computer from the Inside Out
Murray Sargent III i Richard L Shoemaker
Editorial Addison-Wesley
1984

3 IBM DOS 2 00 Manual IBM CORPORATION 1983

4 Managing Memory, A guided tour of DOS 2 0
Memory management
William J Redmond
PC Tech Journal
Volume 2 # 2
August 1984

Interfaces

Patricio A. Navarrete Encina
Quimico. Academico, U. de Chile.

3ª Parte

En la primera parte de este serie de articulos se definieron los conceptos de interfaz y otros necesarios para continuar con el tema. Como se vio, era posible descargar una gran parte del hardware necesario para construir una interfaz derivandolo hacia software adecuado, con la consiguiente baja en la inversion necesaria para realizar el proyecto.

En la segunda parte se aplico lo anteriormente dicho a un problema espacifico de adquisicion de datos directamente utilizando un microcomputador de bajo costo, el cual a pesar de no poseer una compuerta de usuario permite la entrada de información analoga a través de las compuertas usadas habitualmente para conectar bastones de juego y otros. En este caso, practicamente todo el hardware de la interfaz fue substituido por el ya existente en el microcomputador y por software apto para realizar las operaciones necesarias.

Entrada y salida de información

Tanto la entrada como la salida de la información hacia y desde un microcomputador puede ser de dos tipos: binaria o digital y análoga.

Para ciertos microcomputadores esta operación de entrada y salida es realizada facilmente, ya que llevan incorporado el hardware adecuado, restando entonces configurar a traves de software las compuertas o adaptadores de interfaz para la operación a realizar.

De los multiples tipos de microprocesadores usados en la actualidad, restringiremos el análisis al 6502 por ser uno de los mas comunes.

El microprocesador 6502 es usado por microcomputadores tales como Apple II, Atari, Commodore, PET, Vic-20, entre otros y puede tener acceso a 65536 localizaciones de memoria, numeradas de 0 a 65535.

Algunas de las localizaciones de memoria contienen el RAM y es posible alterar el contenido de estas memorias usando instrucciones como POKE (en Basic) o bien leerlas con la instrucción PEEK. Cada localización de memoria contiene un byte (de 8 bits) y por lo tanto sólo se puede almacenar en ella valores entre 0 y 255.

Otra parte de la memoria está dedicada al ROM (Read Only Memory), la cual es usada para guardar el sistema operativo, el intérprete de lenguaje u otros que no puedan ser alterados por los usuarios con instrucciones como POKE, aunque sí pueden ser leidas; de este modo se evita modificaciones accidentales que provocarian la "caida" del sistema. El ROM a diferencia del RAM no es volatil, es decir retiene su programación aunque el computador sea apagado.

El procesador 6502 utiliza dispositivos de entrada y salida que se ubican en ciertas posiciones de la memoria y pueden ser leidos y alterados como si fuesen RAM.

Para cada microcomputador particular estas direcciones varian y se debe consultar las guias de referencia y bibliografia adecuada al microcomputador usado para conocer estas ubicaciones.

Volviendo al tema de los tipos de senales de entrada salida, encontramos que habitualmente los microcomputadores manipulan informacion binaria directamente y pueden aceptar información análoga disenando una interfaz capaz de transformar las señales analogas a binarias. Esta operación la realiza un dispositivo que se llama Convertidor Análogo-Digital (CAD). El proceso inverso de conversion de informacion digital a análoga es realizado por los conversores digital-análogo (CDA).

Los instrumentos que poseen lectura digital a menudo poseen salidas BCD (Binary Coded Decimal), lo cual permite conectarlos a otros dispositivos y eventualmente a un microcomputador, siendo en este caso bastante simple efectuar dicha conexion.

Como ejemplo tomaremos la conexion de un medidor digital de panel con un microcomputador provisto del microprocesador 6502 y el adaptador de interfaces 6522 ó 6526 (este en el caso de los microcomputadores PET, Vic-20 y Commodore 64).

El medidor digital de panel NLS PM-450 es un medidor de panel de 4½ digitos. Es posible construir una interfaz con un solo CI utilizando 8 lineas de datos del adaptador 6522 (llamado VIA) y la linea CB2 de control.

El medidor digital de panel (MDP) posee además de la salida binaria una linea de "mantencion" (hold), la cual una vez activada impide un cambio en los digitos del panel y de las salidas, lo cual es util para poder leer los numeros a una velocidad adecuada. La linea CB2 de la VIA debe ser conectada entonces directamente a la linea de mantención del MDP.

Para poder definir los digitos entre 0 y 9 es necesario que la salida BCD tenga 4 lineas y el MDP posee otras 5 lineas para definir la posición del digito en exhibición. Esto en total hace que el MDP posea 9 lineas de salida: 4 para poder representar cada digito en codigo BCD y 5 para fijar la posicion del mismo. Las lineas funcionan en conjunto de modo que cada digito es exhibido secuencialmente

Continua en pag. 42

a una velocidad tal que es prácticamente imperceptible y el visor aparece como continuamente encendido con todos los dígitos visibles "simultáneamente".

Hay entonces un problema que debemos enfrentar. tenemos 9 salidas por parte del MDP y 8 entradas de la VIA

La resolución del puzzle se logra si nos damos cuenta que las 5 líneas que fijan las posiciones del dígito en exhibición actúen en consonancia con un estado bajo correspondiente a la posición del dígito, o sea, existe para estas 5 líneas solamente 5 estados posibles, los cuales los podemos reducir fácilmente a 4 utilizando por ejemplo 4 puertas NAND de dos entradas (Ver Fig. 1). Como los niveles lógicos de las salidas del MDP y las entradas a la VIA son compatibles con la familia lógica TTL, la conversión la realizaremos utilizando el CI 7400 o su equivalente 74LS00 que es precisamente un CI que contiene 4 NAND de dos entradas.

En la Tabla 1 se detalla la relación existente entre la posición del dígito y sus estados lógicos con los valores obtenidos luego de ser procesados por el CI 7400. Estos últimos valores, S1 a S4, son ingressdos a la VIA

Tabla 1

Posición del Dígito	Estados lógicos					Conversión				Equivalente Decimal
	P4	P5	P3	P2	P1	S4	S3	S2	S1	
1	1	1	1	1	0	0	0	0	1	1
2	1	1	1	0	1	0	0	1	1	3
3	1	1	0	1	1	0	1	1	0	6
4	1	0	1	1	1	1	1	0	0	12
5	0	1	1	1	1	1	0	0	0	8

En la Fig. 1 se observa el diagrama lógico de la interfaz compuesta por un solo CI en la cual se describen las entradas a cada una de las puertas NAND del 7400 y las salidas del mismo. Se tiene que las cinco entradas P1 a P5 han sido reducidas a 4 salidas, lo cual implica que el CI efectúa una conversión que es la descrita en la Tabla 1

P1 P2 P3 P4 P5 — S1 S2 S3 S4

P1 P5 Posición de dígito

S1 S4 Salidas de la interfaz

Figura 1

El conexionado físico del CI 7400 se detalla en la Fig. 2 y finalmente en la Fig. 3 se describe el esquema completo desde el MDP al computador incluyendo la interfaz

P1 1, P2 2, S1 3, 4, P3 5, S2 6, Gnd 7 — 14 Vcc, 13 P5, 12, 11 S4, 10 P4, 9, 8 S3

Figura 2 **Conexiones físicas al CI 7400**

MDP: Hold, D0, O3, P1, P5 — CI S1 – S4 — VIA: CB2, Io, 17

D0 – D3 Líneas BCD

I0 – 17 Entradas a la VIA

S1 – S4 Salidas del CI 7400

Figura 3

Notas sobre el Software

Una vez que la información ha ingresado al computador, el software a emplear depende tanto de las características propias del lenguaje o dialecto Basic que emplea el microcomputador como de las preferencias del programador. Sin embargo algunas ideas pueden ser de utilidad y evitar más de algún dolor de cabeza

En primer lugar el programa debe fijar el registro de direcciones de la VIA para recibir la información y para mantener fijo y liberar luego al MDP una vez leídos los dígitos, luego el programa debe leer el valor almacenado en la memoria seleccionada y separar el byte leído en los dos niveles correspondientes ya que en uno de ellos se encuentra el dígito y en el otro la posición del mismo y fueron leídos simultáneamente

Una manera de realizar esta operación consiste en usar una subrutina en lenguaje de máquina que efectúe un AND lógico entre el valor almacenado y un par de números adecuados para enmascarar los 4 bits más significativos (240) en un caso y los 4 bits menos significativos (15) en el otro

El software debe por lo tanto, almacenar el byte leído y copiarlo en otra memoria, de modo que podamos efectuar las operaciones de enmascaramiento sin perder el valor original

Veamos un ejemplo

Supongamos que el valor almacenado es 76 y efectuemos la operación AND lógico con 240 primero y con 15 después

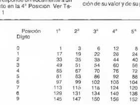
		nibble superior	nibble inferior
240		1 1 1 1	0 0 0 0
76	AND	0 1 0 0	1 1 0 0
		0 1 0 0	0 0 0 0

El valor almacenado en el nibble superior corresponde al dígito 4

	nibble superior	nibble inferior
	0 0 0 0	1 1 1 1
	0 1 0 0	1 1 0 0
	0 0 0 0	1 1 0 0

El valor 12 del nibble inferior corresponde unívocamente a un dígito en la 4ª Posición Ver Tabla 1

Como esto parece algo complicado podemos establecer un algoritmo de trabajo adecuado si estudiamos los valores de la Tabla 2 que corresponden al valor decimal almacenado en función de su valor y de su posición



Posición / Dígito	1ª	2ª	3ª	4ª	5ª
0	1	3	6	12	8
1	17	19	22	28	24
2	33	35	38	44	40
3	49	51	54	60	56
4	65	67	70	76	72
5	81	83	86	92	88
6	97	99	102	108	104
7	113	115	118	124	120
8	129	131	134	140	136
9	145	147	150	156	152

Tabla 2

En la Tabla 2 se puede observar que para cada posición o columna el valor de un numero puede obtenerse sumando 16 al anterior, y, otra observación importante, es que los valores de la 1ª fila (correspondiente al dígito 0) son los mismos que los equivalentes decimales que figuran en la Tabla 1 Entonces de aquí deducimos nuestro algoritmo

a) Dividir al numero almacenado por 16 la parte entera corresponde al dígito

b) Multiplique el resto por 16 El valor obtenido corresponde a uno de los valores de la primera fila de la Tabla 2 y que ya vimos, corresponden unívocamente a los valores de posición del dígito

Veamos un ejemplo

Valor almacenado

a) $\frac{99}{16} = 6\ 1875$

dígito — resto

b) 0 1875 • 16 = 3 (corresponde a un dígito de la 2ª posición)

O sea, el valor almacenado corresponde al dígito 6 en la 2ª posición

_ _ _ 6 _

Conclusion:

El metodo aplicado para conectar el MDP al microcomputador puede ser aplicado en general a cualquier dispositivo que disponga de salida BCD multiplexada En el caso del MDP descrito, que detecte variaciones de potencial, puede ser usado en cualquier sistema físico en el que se produzca este tipo de variaciones y si se usa en combinación con el reloj interno del microcomputador se tiene una excelente y versátil combinacion para un sistema de almacenamiento de información

El software utilizará la informacion y la procesará de acuerdo a los intereses de cada usuario en particular

"La recursión en Pascal"

Julio Falcon Lucero
Sergio Galarce Solas

¿Quién no se ha visto alguna vez entre dos espejos y ha observado su imagen repetida una dentro de otra, hasta el infinito? ¿Habrá alguien que no haya imaginado ver en la pantalla de su televisor otro televisor y en la pantalla de este otro más y mas y mas ? Podría seguir citando muchos más casos de la vida real en los que nos topamos con la recursión, pero

En el presente artículo comenzaremos por definir lo que es un programa recursivo, luego, con algunos ejemplos veremos ventajas y desventajas de este tipo de programas

Todos los algoritmos están codificados en pseudocódigos excepto aquellos en los que se indique otra cosa

DEFINICION Un programa recursivo P, puede ser expresado como un conjunto W de sentencias Si (no conteniendo P) y P mismo

$$P = W[Si\ P]$$

Las herramientas necesarias para expresar programas recursivos son los procedures o subrutinas permitiendo sentencias que puedan tomar un nombre para invocarlos Un procedure que se está llamando a sí mismo, directa o indirectamente se dice recursivo

Si un procedure tiene una referencia explícita a sí mismo, entonces es directamente recursivo

```
PROCEDURE P (lista de parámetros)
BEGIN
        ___
        ___
        CALL P (lista de argumentos)
END
```

Ahora, si la referencia es hacia otro procedure Q, el cual tiene una referencia directa o indirecta a P, se dice entonces que P es indirectamente recursivo

```
PROCEDURE       P (lista de parámetros)
BEGIN___

        CALL    Q (lista de argumentos)
END
PROCEDURE       Q (lista de parametros)
BEGIN___

        CALL    P (lista de argumentos)
END
```

El uso de la recursión por tanto, puede no ser inmediatamente claro desde el texto del programa Permite a menudo una descripción más clara del algoritmo de lo que sería posible sin utilizar la recursión

Los algoritmos recursivos se usan, principalmente cuando el problema a ser resuelto o la estructura de datos a ser procesada, están definidos en términos recursivos como por ejemplo

i) Recorridos de arboles
ii) Exploración y evaluación de expresiones matemáticas
iii) Juegos en los que un movimiento se puede dar en función de otro de menor nivel de complicación

RECURSION E ITERACION Un programa recursivo puede ser escrito casi directamente de la definicion del programa en tanto que la forma no recursiva o iterativa requiere de cierto grado de habilidad para programar Además el programa recursivo resulta más comprensible para otra persona

Existen muchas funciones matemáticas que pueden ser definidas recursivamente, por ejemplo

Factorial (n!) $\begin{cases} 1 & \text{si } n = 0 \\ 1 & \text{si } n = 1 \\ n(n-1)! & \text{si } n > 0 \end{cases}$

Función exponencial(x^n) $\begin{cases} 1 & \text{si } n = 0 \\ x * x^{n-1} & \text{si } n > 0 \end{cases}$

Polinomio de Lagrange $P_n(x)$ $\begin{cases} 1 & \text{si } n = 0 \\ x & \text{si } n = 1 \\ [(2*n-1)*P_{n-1}(x) - (n-1)*P_{n-2}(x)]/n & \end{cases}$

Cálculo de n!

Solución recursiva

```
    FUNCTION FACT (N)
    BEGIN
        IF (N = 0) THEN FACT = 1
                   ELSE FACT = N*FACT (N - 1)
    END
```

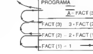
Solución iterativa

```
    FUNCTION FACT (N)
    BEGIN
        FACT = 1
        CONT = 1
        WHILE (CONT< = N) DO
            BEGIN
              FACT = FACT*CONT
              CONT = CONT + 1
            END
    END
```

Ahora veremos el esquema de llamadas recursivas para el cálculo del factorial

Sea el siguiente trozo de un programa

A = FACT (3)



Ocurre a veces que un problema que se podría resolver recursivamente, ocupa mucha memoria debido al tiempo que toma la administración de los stacks, pilas o almacenamiento LIFO (Last In, First Out)

Resulta que, como en el problema anterior el programa tiene que llegar a la solución elemental, es decir FACT (1) = 1, para luego irse devolviendo

FACT (2) = 2 * FACT (1) = 2 * 1 = 2
FACT (3) = 3 * FACT (2) = 3 * 2 = 6 para todo esto se necesita un puntero que indique donde volver

LIFO, como su nombre lo indica, quiere decir que el último elemento en entrar es el primero en salir. En nuestro caso, estos elementos son las direcciones a dónde debe volver el programa

Otro problema que se produce es que, si un programa es erróneo, será difícil encontrar el error y por tanto corregirlo, particularmente si el nivel en la recursión es demasiado profundo. Por lo tanto, se deben pesar los beneficios de un programa sencillo contra el tiempo adicional de ejecución y la posibilidad de trastorno en la corrección de errores

Un caso en que la solución recursiva no es recomendable, es el de la solución de los polinomios de Lagrange

```
Solución recursiva
FUNCTION P (N,X)
BEGIN
        IF(N = 0)  THEN P = 1
                   ELSE IF(N = 1)
                         THEN P = X
                         ELSE   P = ((2*N-1 )*P(N-1 X)-
                         (N-1)*P(N-2,X))/N
END
```

Un ejemplo claro del stack que debería mantener este programa, es el siguiente diagrama de árbol si n = 5

Basta tan sólo con imaginar cuál sería el diagrama para un valor de n>5. Los stacks, para mantener las posiciones dónde regresar harían este programa muy lento además de ocupar mucha memoria

Entonces comprendemos que para este problema, la solución iterativa, a pesar de ser menos clara, es la más eficiente

```
Solución iterativa
FUNCTION P (N,X)
BEGIN
      IF(N = 0)  THEN P = X
                 ELSE
                    BEGIN
                        PREVIO = 1
                        ACTUAL = X
                        I = 2
                        WHILE (I< = N) DO
                        BEGIN
                             SIGTE = (2*I - 1)*
                             ACTUAL-(I - 1) *
                             PREVIO)/I
                             PREVIO = ACTUAL
                             I = I + 1
                        END
                        P = SIGTE
                    END
END
```

Es claro que deben adoptarse medidas especiales para este tipo de procedures, ya que necesitan una jerarquía completa de enlaces y registros para espacio de trabajo, de manera tal que el estado de la subrutina en su activación inmediatamente precedente, pueda ser restaurado al término de la activación actual

En aplicaciones prácticas es necesario indicar que LA PROFUNDIDAD EN LA RECURSION NO DEBE SER SOLAMENTE FINITA, SINO QUE DEBE SER PEQUEÑA

Las Torres de Hanoi

A continuación, y para concluir, presentaremos un algoritmo que imprima ordenadamente los movimientos que se deben realizar para resolver el problema de las Torres de Hanoi

El objetivo de este juego es mover N discos desde la torre 1 a la torre 3, uno a uno y de forma tal que, en ningún momento, quede colocado un disco sobre otro de tamaño inferior, para lo que se puede emplear como torre de almacenamiento temporal a la torre 2

Sea N la cantidad de discos a mover. Mediante el algoritmo recursivo ["Programación con el lenguaje Pascal", F.J. Sanchis y A. Morales, pág. 271], se concluye que la cantidad de movimientos necesarios para mover N discos desde 1 a 3, es de:

$$\text{Cantidad de movimientos} = 2^N - 1$$

Solución para el caso de N discos:

```
Procedimiento general
    MOVER (N, 1, 2, 3) La tarea se puede llevar a cabo
en las siguientes subtareas:
    MOVER (N - 1, 1, 3, 2)
    MOVER (N - 1, 3, 2, 1)
  Solución generalizada
    BEGIN
        READ (TOTAL)
        CALL MOVER (TOTAL, 1, 2, 3)
    END

PROCEDURE MOVER (NUMERO, DESDE, HACIA,
TEMPO)
BEGIN
  IF(N>0)
    THEN
    BEGIN
      CALL MOVER (NUMERO - 1, DESDE, TEMPO,
      HACIA)
      WRITE (DESDE, <-> , HACIA)
      CALL MOVER (NUMERD - 1, TEMPD HACIA,
      DESDE)
          END
    END
```

Torre 1 Torre 2 Torre 3

Podemos presentar la siguiente implementación al lenguaje LOGO

```
CREAR THANDI TOTAL
MOVER TOTAL 1 2 3
FIN
CREAR MDVER NUMERO DESDE TEMPO HACIA
SI NUMERO > D [CICLO]
FIN

CREAR CICLO
MOVER NUMERO - 1 DESDE HACIA TEMPO
(IMPRIMA (.DESDE .HACIA))
MDVER NUMERO - 1 TEMPD DESDE HACIA
FIN
```

Este sencillo programa nos muestra los movimientos a realizar para dejar todos los discos en la torre 2

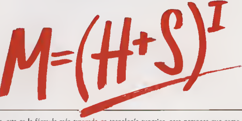
Bibliografía:

"Algorithms + Data Structures = Programs" N Wirth

"Programación con el lenguaje Pascal" F J Sanchis, A Morales

"Recursive Techniques in Programming" D.W Barron

LA FORMULA PERFECTA



Sin duda, esta es la fórmula más avanzada en tecnología superior, para personas que como usted, lideran la utilización de la información en su empresa. Y las razones sobran ¿Por qué? Despejemos la incógnita:

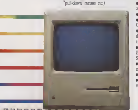
M(Macintosh)=(H(Hardware)+S(Software)) Inteligencia

El computador personal mas avanzado poderoso sofisticado y fácil de usar

- Arquitectura de 32 – bit
- Pantalla de alta resolución (512 x 324 pixel)
- 64 K ROM y RAM de 128 a 1MB Reloj interno de 7.83 MH
- Tamaño compacto y portable
- Interfases intuitivas de alta productividad (Mouse, iconos ventanas "pull-down" menus etc.)

- Multiplicidad de lenguajes de alto nivel = Pascal, "C" Fortran Forth Lisp Modula 2, etc
- Emulación de terminales IBM, DEC/VAX, Hewlett Packard, Data General, entre otros
- Paquetes Estadísticos
- Software para manejo de bibliotecas de aplicaciones y documentación
- Control de proyectos Pert/CPM y otros.
- Poderosas bases de datos Omnis 3, Odesta, Helix etc
- Software CAD que permite desde documentar SIA a realizar diseños de circuitos
- Paquetes integrados Jazz de Lotus, Excel de Microsoft, Quartet de Haba Systems.
- Accesorios para digitalizar documentos e imágenes
- Ram disk.
- Switcher para integrar en memoria multiples aplicaciones.

Toda la inteligencia del Macintosh elevada al servicio de su trabajo



apple

en Chile, con el respaldo de XEROX

Distribuidores autorizados Iquique COMPUTRONICA, Manzana D Sitio 40 D Fono 23103 Antofagasta CONDE ANTOFAGASTA, Av B. O'Higgins 1912 Dpto 205 Fono 226057 / COMERCIAL GALICIA, Baquedano 733 Fono 222775 Santiago ADMINISTRACION E INFORMATICA LTDA., Nueva York 80 10° Piso Fono 6960052 / COMPUTERLAND La Concepción 80 Fono 2239512 / INTERNATIONAL DATA SERVICES Mac Iver 115 local 9 Fono 392500 PRODAGOM LTDA. Av Andres Bello 1581 Fono 2232087 TEOREMA COMPUTADORES Parque Arauco Local 347 A, Fono 2420996 Aguatinta 1160 Fono 722291 Viña del Mar XEROX LTDA. Av Ecuador 17 Fono 81652 TALCA SAINO LTDA. 2 Oriente 1141 Fono 35555 Concepción MICROCOMPUTADORES Y COMUNICACIONES LTDA. O'Higgins 770 Local 14, Fono 29222. Temuco FIRMANI Y COMPAÑIA LTDA. Manuel Montt 730 Fono 34239 Osorno FIRMANI Y COMPAÑIA LTDA., Ramirez 870 Fono 4566 Punta Arenas AUTOMOTORES SADEM LTDA. C.P.A. Balmaceda 833 Fono 25090

Programación de trabajos y máquinas

Guillermo Beuchat
Ing. Civil Industrial U. de Chile.



La secuenciación de trabajos en una serie de máquinas es un problema clásico en la Administración de Operaciones, para el cual se han desarrollado numerosas heurísticas que pretenden minimizar el tiempo total de entregas de trabajos o minimizar el tiempo ocioso de cada máquina. Como parte de nuestra serie de artículos sobre aplicaciones de la microcomputación a la Ingeniería Industrial, presentamos en esta oportunidad un programa Basic que calcula la secuencia óptima de trabajos para un numero fijo de máquinas o centros de proceso haciendo uso de una heurística desarrollada por L. F. Gelders y N. Sambandam (1).

Antes de abordar el uso del programa en situaciones reales, es necesario definir algunos conceptos básicos relacionados con la teoría de programación de máquinas. En primer lugar, ¿en qué consiste este proceso y cuál es la justificación del uso de métodos matemáticos para su solución?

Reid y Stark (2) aportan una definición adecuada de la programación de trabajos y máquinas. Se sostiene que los trabajos representan órdenes o pedidos de los clientes, y las máquinas o estaciones de trabajo representan los medios para satisfacer los pedidos. La secuencia que deben seguir todos los trabajos al pasar por las máquinas se denomina secuencia técnica y absolutamente todos los trabajos deben seguir la misma secuencia técnica para pasar por las máquinas.

El problema consiste entonces en ingresar las órdenes o trabajos a la secuencia productiva de tal forma de optimizar algún parámetro del sistema, tal como el tiempo ocioso de las máquinas o el tiempo de entrega de los productos. Estos tiempos representan por supuesto un costo para la empresa, por lo que la determinación de una secuencia óptima de ingreso de trabajos a las máquinas adquiere relevancia económica, especialmente cuando el numero de trabajos es grande.

Para entender mejor el concepto de secuenciación supongamos el siguiente ejemplo: una imprenta recibe dos trabajos diferentes, cada uno de los cuales requiere pasar por dos máquinas: la prensa offset y la guillotina, en ese orden. El primer trabajo (T1) requiere 4 horas de prensa y 1 5 de guillotina, mientras que el segundo (T2) requiere solamente 1 hora de prensa y 3 5 de guillotina. El administrador desea saber cuál de los dos trabajos debe efectuar primero, suponiendo que ambos se reciben en el taller al mismo tiempo y lo que se desea es minimizar el tiempo ocioso total de las máquinas. Analicemos los dos casos.

a) procesar primero T1 y luego T2.

En este caso, ingresa T1 a la prensa, lo que implica de inmediato 4 horas de tiempo ocioso en la guillotina. Luego, T1 pasa a la guillotina, y simultáneamente T2 ingresa a la prensa. Dado que T2 requiere 1 hora de prensa, el trabajo tendrá 0 5 horas de espera hasta que se desocupe la guillotina, ocasionando 0 5 horas de tiempo ocioso a la prensa. Finalmente, T1 sale de la guillotina y T2 entra, sumando 3 5 horas de tiempo ocioso a la prensa. La siguiente tabla muestra los tiempos ociosos para cada máquina y total:

	Prensa	Guillotina	Total
T1		4	4
T2	0 5 + 3 5		4
			8

b) procesar primero T2 y luego T1.

En este caso, ingresa T2 a la prensa, ocasionando de inmediato 1 hora de tiempo ocioso a la guillotina. Luego, T2 pasa a la guillotina, y simultáneamente T1 ingresa a la prensa. Dado que T2 requiere 3 5 horas de guillotina y T1 requiere 4 horas de prensa, la guillotina quedará libre durante 0 5 horas. Finalmente, T2 sale de la guillotina y entra T1, lo que implica 1 5 horas ociosas de la prensa. La siguiente tabla muestra los tiempos ociosos para cada máquina y total:

	Prensa	Guillotina	Total
T1	1 5	0 5	2
T2		1	1
			3

Continúa en pág. 50



para Coasin nada es imposible

Porque somos una empresa de alto nivel tecnológico especializados en el área de telecomunicaciones, respaldado por un Departamento de Ingeniería integrado por calificados profesionales chilenos, equipados con moderno instrumental para controlar el óptimo funcionamiento de cada equipo. De este modo, asesoramos y aportamos soluciones a las necesidades de su empresa.

- Comunicación de datos
- Sistemas Periféricos "directamente" compatibles con IBM
- Automatización de oficinas
- Sistemas de protección eléctrica y acondicionamiento de líneas.

Coasin

Santiago

Obviamente, el tiempo ocioso total de 3 horas es mucho menor que la alternativa anterior. La conclusión es que el orden óptimo en que deben procesarse los trabajos es T2 – T1.

El ejemplo anterior nos permite apreciar también que el problema de cálculo se vuelve altamente complejo a medida que crece el número de máquinas y trabajos. Por otra parte, el criterio de optimización usado no es el único, ya que es posible definir otros de acuerdo a la función objetivo de cada empresa en particular.

Trabajos con prioridades diferentes

El caso de la imprenta lleva implícito una suposición muy importante: que todos los trabajos tienen igual prioridad para ingresar al proceso. Sin embargo, ello no es lo común en situaciones reales, en que algunos trabajos de clientes importantes deben ser procesados en forma urgente, o bien la empresa incurre en costos o multas por entregas atrasadas del producto. Otra razón por la que un trabajo en particular debe tener prioridad se produce cuando las materias primas que utiliza son perecibles, es decir, se incurre en un costo de almacenamiento asociado a la demora en ingresar *un trabajo al proceso productivo. Todas estas consideraciones* hacen necesario el desarrollo de heurísticas que tomen en cuenta situaciones reales y que puedan entregar una secuencia óptima cuando se presentan las distorsiones analizadas.

El modelo de Gelders y Sambandam

La heurística desarrollada por estos autores es la base del programa Basic que se presenta a continuación, que permite resolver el problema de secuenciación para N trabajos en M máquinas. El método ha sido probado en numerosos problemas reales y se ha encontrado que proporciona muy buenas soluciones, las que son óptimas para el caso simple de dos máquinas.

Los supuestos básicos del modelo son los siguientes:

a) se toman en cuenta costos por atraso en la entrega de productos, a través de la especificación de fechas de entrega para cada trabajo que fijan la prioridad de cada uno
b) se toman en cuenta costos por demora en el ingreso de trabajos al sistema, asociados a costos de almacenamiento u otros
c) los tiempos requeridos por cada trabajo en cada máquina se suponen conocidos
d) todos los trabajos están simultáneamente disponibles al momento de hacer el análisis
e) todos los trabajos deben pasar por todas las máquinas en forma secuencial, aunque pueden existir máquinas con tiempo cero para algún trabajo

La heurística se basa en dos principios fundamentales: dar prioridad a los trabajos más 'caros' en cuanto a la demora o atraso y minimizar luego el tiempo ocioso total de las máquinas. Para obtener un resultado, el programa requiere de los siguientes datos de entrada:

a) el número de trabajos y de máquinas
b) los tiempos de proceso de cada trabajo en cada máquina
c) los tiempos de entrega contratados
d) los costos por atrasos
e) los costos por demora

El modelo entrega como salida la mejor secuencia calculada y el costo total asociado. Sin embargo, es posible ampliar estas salidas para incluir, por ejemplo, una indicación de las fechas reales de entrega de cada trabajo y los costos de demora o atraso de cada uno. Las modificaciones son simples y su implementación podrá ser realizada por los lectores interesados.

Uso del programa Basic

Mediante un ejemplo real, analizaremos la manera de utilizar el programa propuesto para obtener una secuencia óptima de procesamiento para 5 trabajos en 4 máquinas. El programa se basa en una versión desarrollada por Woollam y Sambandam (3).

Juan Moya, administrador de la pastelería La Crema Ltda., acaba de recibir 5 pedidos de diferentes *clientes, todos con bastante urgencia.* La siguiente tabla muestra el resumen de los trabajos encargados, incluyendo el tiempo de entrega y la descripción de cada uno.

Trabajo	Descripción	Tiempo entrega (horas)
1	250 empanadas pino	25
2	220 empanadas queso	30
3	16 doc. pasteles crema	40
4	12 doc. tartaletas	51
5	1 torta novia	55

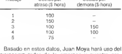
Por otra parte, después de un estudio de los pedidos, se ha estimado que los requerimientos de tiempo (en horas) de cada pedido en cada máquina, incluyendo entre éstas al tiempo necesario para el trabajo del maestro pastelero, son los siguientes:

Trabajo	Amasadora	Pastelero	Cocina	Horno
1	8	2	9	2
2	10	1	7	2
3	8	12	4	8
4	10	1	9	13
5	8	10	1	2

Dado que los clientes son todos muy importantes, Juan Moya cree que todos los pedidos implican costos por atraso bastante altos, mientras que solamente los trabajos 3 y 4 tienen costos de almacenamiento asociados al uso del refrigerador para las materias primas de los pasteles de crema y las tartaletas. La siguiente tabla muestra los costos asociados a cada trabajo.

Continúa en pág. 53


BYTESHOP
PRECIOS IMBATIBLES
COMPUTADORES SPECTRUM
16 K .... $ 29 900
48 K .... $ 39.900
Software
Interfaces
Accesorios
IMPRESORAS
STX-80 (60 cps) US$ 396
Gemini 10 (120 cps) US$ 594 *
Gemini 15 (120 cps) US$ 912 *
Smith Corona F-80 US$ 420 *
Gran Oferta
Seikosha GP 505 .... $ 39 900
Compatible ZX-81 y Spectrum
SUPEROFERTA DISKETTES
Xidex
SSDD .... $ 580
DSDD .... $ 740
3,5 .... $ 1.150
Verbatim 5¼"
SSDD .... $ 516
DSDD .... $ 680
Verbatim 8"
1 D .... $ 860
2 D .... $ 1.005
Nuevos diskettes Certron 5¼
SSDD .... $ 490
DSDD .... $ 590
Porta diskette (10) .. $ 950
¡¡Ojo!!
Por la compra de diez diskettes Certron, exija gratis un porta diskettes
FELIZ NAVIDAD
COMPUTADORES SPECTRUM
16 K .... $ 29 900
48 K .... $ 39 900
Software
Interfaces
Accesorios
ATENCION PROVINCIAS ¡¡DESPACHAMOS CONTRA REEMBOLSO EN 48 HRS !!
Alameda 108 – Local 204 – Fono 399051
Galería Hotel Crowne Plaza
*Todos los precios incluyen IVA
**Oferta válida hasta agotar stock
***Byteshop se reserva el derecho de modificar los precios.
* Equivalente moneda nacional



Trabajo	Costos por atraso ($/hora)	Costos por demora ($/hora)
1	100	–
2	150	–
3	100	150
4	100	100
5	75	–

Basado en estos datos, Juan Moya hará uso del microcomputador de la empresa y nuestro programa Basic para resolver el problema

Para ello es necesario ingresar los datos expuestos en líneas Data, tal como se muestra en el listado Por último se hace correr el programa y se digita el número de trabajos y el número de máquinas para obtener el resultado que se muestra

Tras un breve tiempo para efectuar los cálculos, el programa entrega una lista de las secuencias que va analizando hasta que finalmente calcula la mejor solución para el problema y entrega la secuencia de ingreso de trabajos

Orden	Trabajo	Descripción
1	3	16 doc pasteles crema
2	4	12 doc tarlaletas
3	2	220 empanadas queso
4	1	250 empanadas pino
5	5	1 torta novia

Este ordenamiento que minimiza el tiempo ocioso total tiene un costo de 13 700 pesos Las modificaciones propuestas al programa permitirían conocer cómo se desglosa este costo, y por qué concepto se produce en cada trabajo

Conclusion

La heurística que hemos presentado aquí es una de muchas que existen para resolver el problema de secuenciación de trabajos Es importante destacar que no todas son aptas para resolver todos los problemas dados los supuestos simplificatorios que se hacen Por ello, este es un caso que siendo específico, se puede adaptar a una gran cantidad de situaciones reales Un ejemplo de esto es que alguno de los trabajos no necesite pasar por alguna de las máquinas En este caso bastaría con indicar un tiempo de proceso igual a cero y se supera la restricción de la heurística que supone que todos los trabajos deben seguir la misma secuencia tecnológica

Como siempre queremos hacer énfasis en el hecho de que las empresas nacionales y sus ejecutivos requieren urgentemente de un cambio radical en el enfoque de gestión y administración

Dada la creciente complejidad y competitividad de los mercados se hace necesario incorporar el uso de técnicas modernas de gestión Esta serie de artículos ha demostrado y seguirá demostrando que es posible cumplir ese objetivo a bajo costo y con un mínimo de dedicación, maximizando el rendimiento del personal y de los recursos materiales y financieros de la empresa

Referencias

(1) FOUR SIMPLE HEURISTICS FOR SCHEDULING A FLOW-SHOP L F GELDERS Y N SAMBANDAM
International Journal of Production Research, Vol 16 # 3, 1978

(2) JOB SEQUENCING PROGRAM MINIMIZES PROCESSING TIME
R A REID Y W A STARK
Industrial Engineering, January 1982

(3) FLOW SHOP SEQUENCE SOLVED FOR HOLDING TARDINESS COSTS C R WOOLLAM Y N SAMBANDAM
Industnal Engineenng, June 1984


```
10 PEM ********************************
20 REM *                              *
30 REM *  PROGRAMACION DE N TRABAJOS  *
40 REM *       EN M MAQUINAS          *
50 PEM *                              *
60 REM *  HEURISTICA DE L.F. GELDERS  *
70 PEM *       Y N. SAMBANDAM         *
80 REM *                              *
90 REM *  GUILLERMO DEUCHAT     1985  *
100 PEM*                              *
110 PEM********************************
120
130 BP$=CHR$(147)
140 FRINT BP$ PEM BOFRA PANTALLA
150 FRINT "PROGPAMACION DE MAQUINAS'
160 PRINT "========================"
170 PRINT
180 INFUT "INGFESE NUMEPO DE TRABAJOS ";N
190 INFUT "INGPESE NUMERO DE MAQUINAS ",M
195 FRINT
200 DIM P(N M),D(N),W(N) H(N),S(N)
210 DIM N1(N) N2(N),T9(M) T(M)
220
230 REM ++++ LEER TIEMPOS DE OPEPACION
240
250 FOR I=1 TO N
320 FOR J=1 TO M
340 READ P(I J)
350 NEXT J
370 NEXT I
380
390 PEM ++++ LEER TIEMPOS ENTREGA
400
420 FOR I=1 TO N
440 READ D(I
450 NEXT I
460
470 REM ++++ LEER COSTOS DE ATPASO
480
500 FOR I=1 TO N
520 PEAD W(I)
530 NEXT I
540
550 PEM ++++ LEER COSTOS POP DEMORA
560
590 FOR I=1 TO N
600 PEAD H(I)
610 NEXT I
620
630 PPINT "****** ESPERE ******
640
650 REM ++++ CALCULA SECUENCIA INICIAL
660
670 T1=0 T2=0
680 FOP I=1 TO N
690 S(I)=0 N1(I)=0 N2(I)=I
700 NEXT I
710 FOR I=1 TO M
720 T(I)=0
730 NEXT I
```

```
740 N9=N-1
750 Z=0
760 FOR X=1 TO N9
770 R9=-9999
780 FOR I=1 TO N
790 IF N2(I)=0 THEN 1030
800 J9=N2(I)
810 FOR J=1 TO M
820 T9(J)=T(J)
830 NEXT J
840 GOSUB 2090
850 I9=0
860 FOR J=2 TO M
870 I9=I9+(T9(J)-T(J)-P(J9,J))
880 NEXT J
890 IF I9=0 THEN 910
900 GOTO 920
910 I9=1
920 T2=T9(M)+T1-P(J9,M)
930 X1=0
940 IF T2<=D(J9) THEN 970
950 X1=T2-D(J9)
960 X1=X1/(T1-P(J9,M))
970 R1=X1*W(J9)+H(J9)
980 R1=R1/(T9(M)-T(M))
990 R2=R1/I9
1000 IF R9>=R2 THEN 1030
1010 J8=I9
1020 R9=R2
1030 NEXT I
1040 A2=J8
1050 GOSUB 1900
1060 T1=T1-P(J8,M)
1070 N2(J8)=0
1080 N1(X)=J8
1090 S(X)=J8
1100 C=T(M)-D(J8)
1110 IF C>=0 THEN 1130
1120 C=0
1130 Z=Z+W(J8)*C+H(J8)*T(M)
1140 NEXT X
1150 FOR I=1 TO N
1160 IF N2(I)=0 THEN 1190
1170 J9=N2(I)
1180 I=N
1190 NEXT I
1200 N1(N)=J9
1210 S(N)=J9
1220 A2=J9
1230 GOSUB 1900
1240 C=T(M)-D(J9)
1250 IF C>=0 THEN 1270
1260 C=0
1270 Z=Z+W(J9)*C+H(J9)*T(M)
1280
1290 REM ++++ IMPRIME SECUENCIA INICIAL
1300
1310 PRINT BP$
1320 PRINT "COSTO", "SECUENCIA DE TRABAJOS"
1330 PRINT "-----" "---------------------"
1335 PRINT Z,
1340 GOSUB 2020
1350 PRINT
1380
1390 REM ++++ ITERACION SOBRE LA SECUENCIA
1400
1410 C1=0
1420 FOR I=1 TO N9
1430 U1=N1(I)
1440 N1(I)=N1(I+1)
1450 N1(I+1)=U1
1460 FOR J=1 TO M
1470 T(J)=0
1480 NEXT J
1490 Z1=0
1500 FOR X=1 TO N
1510 J9=N1(X)
1520 A2=J9
1530 GOSUB 1900
1540 C=T(M)-D(J9)
1550 IF C>0 THEN 1570
1560 C=0
1570 Z1=Z1+W(J9)*C+H(J9)*T(M)
1580 NEXT X
1590 IF Z<=Z1 THEN 1660
1600 Z=Z1
1610 FOR X=1 TO N
1620 S(X)=N1(X)
1630 NEXT X
1640 C1=C1+1
1660 U1=N1(I)
1670 N1(I)=N1(I+1)
1680 N1(I+1)=U1
1690 NEXT I
1700 IF C1>0 THEN 1800
1710 PRINT
1715 PRINT
1720 PRINT "** LA MEJOR SECUENCIA ES **"
1730 PRINT
1735 PRINT Z,
1740 GOSUB 2020
1760 END
1770
1800 REM ++++ MUESTRA SECUENCIA INTERMEDIA
1810 PRINT Z,
1820 GOSUB 2020
1830 GOTO 1390
1900 REM ++++ SUBRUTINA CALCULO T
1910
1920 T(1)=T(1)+P(A2,1)
1930 FOR J3=2 TO M
1940 IF T(J3)>T(J3-1) THEN 1960
1950 IF T(J3)<=T(J3-1) THEN 1980
1960 T(J3)=T(J3)+P(A2,J3)
1970 GOTO 1990
1980 T(J3)=T(J3-1)+P(A2,J3)
1990 NEXT J3
2000 RETURN
2010
2020 REM ++++ SUBRUTINA MUESTRA SECUENCIA
2030
2040 FOR K=1 TO N
2050 PRINT S(K)
2060 NEXT K
2070 RETURN
2080
2090 REM ++++ SUBRUTINA CALCULO T5
2100
2110 T9(1)=T9(1)+P(J9,1)
2120 FOR J3=2 TO M
2130 IF T9(J3)>T9(J3-1) THEN 2150
2140 IF T9(J3)<=T9(J3-1) THEN 2170
2150 T9(J3)=T9(J3)+P(J9,J3)
2160 GOTO 2180
2170 T9(J3)=T9(J3-1)+P(J9,J3)
2180 NEXT J3
2190 RETURN
2200
3000 REM ++++ TIEMPOS DE OPERACION
3005
3010 DATA 8 2 9,2
3020 DATA 10 1,7 2
3030 DATA 6 12,4,8
3040 DATA 18,1,9,13
3050 DATA 8 10,1,2
3055
3060 REM ++++ TIEMPOS ENTREGA
3065
3070 DATA 25 30 40 51,35
3075
3080 REM ++++ COSTOS ATRASO
3085
3090 DATA 100,150 100 100 75
3095
3100 REM ++++ COSTOS DEMORA
3105
3110 DATA 0 0,150 100 0

READY.
```

```
PROGRAMACION DE MAQUINAS
========================

INGRESE NUMERO DE TRABAJOS ? 5
INGRESE NUMERO DE MAQUINAS ? 4

****** ESPERE ******

COSTO   SECUENCIA DE TRABAJOS
-----   ---------------------
 13850   3  4  1  2  5
 13700   3  4  2  1  5

** LA MEJOR SECUENCIA ES **

 13700   3  4  2  1  5
```

¿Q.P.P.E.?

Humberto Molina Yons
Diseñador Ecom

Las dudas al respecto son varias y es por ello que me he permitido escribir este artículo con el fin de invitarlos a hacer una reflexion en base al uso que se está haciendo de esta técnica en nuestro trabajo diario.

A través de los años, mediante cursos seminarios y trabajos realizados con programadores con y sin experiencia he podido comprobar que la razón comun por la cual no se usa esta tecnica es simplemente porque no se conoce.

A continuación se plantean 10 preguntas con el fin de entregar los conceptos básicos de la Programación Estructurada.

1. ¿Qué es la Programación Estructurada?
   Es una técnica para confeccionar buenos programas en base a un grupo de reglas más o menos rígidas.

2. ¿Qué es un buen programa?
   Un buen programa debe ser correcto, comprensible, facil de cambiar, escrito eficientemente y de óptima ejecución.

3. ¿Qué ventajas tiene la P.E.?
   Disminuye los problemas de prueba y mantencion de los programas.
   Aumenta la productividad de los programadores.

4. ¿En que se basa la P.E.?
   Se basa en – En Diseño TOP-DOWN
   – La Programación Modular
   – 3 Estructuras Basicas

5. ¿Qué es el Diseño TOP-DOWN?
   Es un proceso ordenado por el cual se organiza la solución de un problema de programación.
   Convierte un gran problema en pequeños problemas fáciles de solucionar independientemente.

6. ¿En que se apoya el Diseño TOP-DOWN?
   En los diagramas de estructura en los cuales se grafican los niveles y módulos definidos por el diseño (HIPO diagramas).

Nivel 0

Nivel 1

Nivel 2

7. ¿Qué es la Programacion Modular?
   Un programa es MODULAR cuando tiene módulos relativamente pequeños (definidos en el diseño), que son independientes entre si y tienen una sola entrada y una sola salida.

8. ¿Qué ventajas tiene la Programación Modular?
   – Es mas fácil de escribir y cambiar.
   – Es mas fácil distribuir y controlar el trabajo.

9. ¿Cuales son las 3 Estructuras Básicas?

a) De Secuencia

a) Presenta la ejecución de una o mas instrucciones o módulos en forma secuencial.
Su ejecución no está condicionada a nada.
Ej.

```
ADD 1 TO CONTADOR
MOVE 0 TO SWITCH
```

b) De Decisión

b) Representa una decisión en la ejecución del programa.
Se evalua una condición.
Si la condición es verdadera se cumple una accion determinada, si la condición es false se cumple otra acción determinada.
Ej.

```
IF condición 1 THEN acción 1
ELSE acción 2
```

c) Ejecute-Mientras

-) Representa la ejecución de un numero determinado de veces de un grupo de instrucciones o módulo dependiendo de una condición (LOOP)

Ej

PERFORM módulo UNTIL condición

10 ¿Existen otras Estructuras? Sí hay 3 estructuras más las cuales son una variación de las 3 basicas

a) DECISION SIMPLE

Similar a la estructura de DECISION pero la acción se cumple solamente si la condición es Verdadera

Ej

IF condición THEN acción

b) DECISION MULTIPLE O CASOS

Representa la evaluación de una condición con mas de 2 alternativas

Ej

IF cond 1 THEN acción 1

IF cond 2 THEN acción-2

ELSE

IF cond 3 THEN acción 3

ELSE acción-4

c) EJECUTE-HASTA

Similar a la estructura EJECUTE MIENTRAS sólo que en este caso la acción indicada se ejecuta a lo menos una vez en el otro caso puede no ejecutarse nunca



En base a estas 10 preguntas, espero haber motivado al lector con el fin de hacer un análisis profesional del tema en cuestión

¿EXISTE UNA TERMINDLOGIA PRECISA?

Señor Director

En atención a los excelentes articulos que publican soy un antiguo y asiduo lactor de MICROBYTE En ocasion de la publicación del mes de noviembre 1985 he decidido manifestarle por escnto mis más sinceras felicitaciones y deseo hacerlo muy especialmente por los articulos MS DOS un Standard de Facto de Luciano Chiang S y por la sene 'Interfaces para Instrumentación', de Patricio Navarrete E , los cuales prestigian una revista nacional de esta naturaleza

El primero de ellos es una excelente visión de lo que es actualmente el mundo competitivo de la computacion Es también un muy buen reflejo de como este acelerado desarrollo tecnológico transforma nuestro lenguaje en una torre de Babel En efecto, Chiang traduce o ¿propone? HARDWARE como MECAMATICA, por constituir las partes ¿mecanicas ? de una ciencia Informática Conozco la proposicion dada por el Dr Enrique Cansado de UTILERIA, (CONIED L A mayo 1985) por ser los utensilios de los cuales se vale la computación No seria más simple conservar COMPUTADOR y PERIFERICDS ?

Al SOFTWARE lo denomina INFORMATICA quizas en mi opinión su mejor definición sigue siendo PROGRAMAS El mismo Dr Cansado propone MENTALERIA asimilándolo al de utileria pero ahora haciendo referencia a la participación de la mente del individuo dedicado a la computación

Finalmente Chiang para el enlace entre Hardware y Software, senala el nombre de SISTEMA OPERATIVO En ingles se lo conoce como FIRMWARE por ser programas generalmente grabados en ROM (también los hay en disco), y que la mayoría de los usuarios los emplean sin conocer de su existencia ¿Por qué no conservar PROGRAMAS DEL SISTEMA OPERATIVD ?

Respecto a la serie Interfaces para Instrumentación , estos artículos constituyen la apertura de una nueva ventana al uso del computador para aquellos que, conociendo la potencialidad de estos equipos no han tenido la posibilidad de una explicacion tan clara pedagogica y apropiada como la que da el senor Navarrete

Felicitaciones nuevamente a la revista y a sus autores

Saluda muy Atte a Ud

Samuel Trumper R

Agradecemos sus conceptuosas palabras En relación a los alcances respecto a algunos terminos, consideramos que las ciencias de la computacion son aun muy jóvenes como para que se hayan decantado terminos precisos para algunos conceptos

Sus proposiciones o las de Chiang sólo las calificará el tiempo

CRITICA CONSTRUCTIVA

Sr Director

Como profesional de la informática e incidentalmente como docente en asignaturas del área, el informarse oportuna y verazmente pasa a ser una necesidad imperiosa, particularmente en un campo tan dinamico como lo es el computacional Por esta razon, soy asiduo lector de su revista desde el primer numero Aunque debo confesar que al principio la vi con algunas reservas (quizás como una aventura editorial mas) he visto còmo ha ido creciendo cuantitativa y cualitativamente, hasta transformarse en un importante medio de divulgación de lo que hoy dia es la informatica

Sin embargo y con el animo de hacer criticas constructivas he observado que la revista está fuertemente inclinada hacia el área de microcomputadores y el uso del lenguaje Basic El Pascal Cobol y Fortran (por mencionar los mas tradicionales) prácticamente no aparecen y a ratos da la impresión (sobre todo a los neófitos en el tema) de que sólo existiera el Basic ¿Obedeca esto a alguna politica editorial o es solo mera coincidencia?

Por ultimo ¿seria posible que publicaran un indice analitico exhaustivo (como en Microbyte N° 12) con mayor frecuencia? Es ideal para ingresarlo a una Base de Datos o Administrador Generalizado de Archivos y disponer asi de un excelente material de consulta

Con mis deseos de que continuen por la misma senda de superación saluda atte a Uds

Manual Sotomayor Alvarez
Ingeniero en Computación e Informatica
Casilla 1820 – Concepcion

Efectivamente ha sido politica de Microbyte abocarse fundamentalmente al Basic debido a que sin ser el mejor es el mas difundido y conocido

No obstante reconocemos que hemos exagerado en la aplicación de esta politica convirtiendo en muy valida la critica que nos hace

Al respecto podemos anunciar que ya está en preparacion una pequeña sene de articulos de introducción y aplicaciones en Pascal para luego seguir con C

El indice analitico por razones de exhaustividad lo publicaremos como es habitual en el primer numero del volumen III

SIGUEN LOS PALITOS

Sr Director

A propósito del desafio planteado en el numero 14 de su prestigiosa revista, me permito enviarle un programa conteniendo una estrategia que modestamente he calificado de infalible para el juego de palitos Este programa esta desarrollado en lenguaje BASIC para la calculadora programable CASIO FX-750 P

Sobre esta calculadora quiero senalar que tiene todo lo necesario como para agregarle una tecla más al lado izquierdo de la tecla ' y por sobre la tecla F (segunda función) Con esta tecla se logra un acceso directo a los caracteres internacionales presionando las teclas correspondientes, sin tener que tipear CHR$(), significando un ahorro de a lo menos 6 pasos de programa lo que para los que no poseemos una maquina de gran memoria es de gran valor Si algun dueno de esta calculadora se interesara por hacerle este agregado, gustoso le entregaré mayores detalles si los necesitase para la instalación de esta tecla La tecla que emplee para este injerto fue una de las teclas de funciones trigonométricas de la calculadora CASIO FX-21, de las que tengo 14 para entregarselas a quien las necesite y no consiga otras similares

Por ultimo, debo hacer una critica acerca de la seccion programas

En el mismo numero 14, el programa del juego de palitos es prácticamente ilegible ocupando además una página completa, lo que a mi entender es excesivo

Sobre lo mismo en el numero 13, se emplea 2 páginas para explicar la conversion de escalas de temperaturas Todo ese espacio me parece util e interesante si se lo dedicase a entregar más programes en lugar de extenderse tanto sobre un tema que no lo requiere

Esperando una buena acogida de la presente y de los programas adjuntos me despido agradeciendole el que me haya permitido reemplazar las revistas extranjeras sobre computación por una nacional y que para orgullo de todos es de un nivel extraordinario a pesar de su corta vida

Karl – R Muller
Casilla 662
Valparaiso

Estimado lector

Nuestra revista trata de reflejar las inquietudes de Uds los lectores Destinamos algo más de espacio a los palitos , publicando su programa

"Estrategia infalible para el juego de palitos"

```
 5 CLEAR INPUT  MAXIMO A SACAR
   MA  X   MA   1
10 A   INT (RND - 5   MA)   1"  WAIT 20
15 PRINT C      B      A   INPUT SA  IF SA
     0 THEN BEEP 1   GO TO 15
   20 IF SA>MA THEN 80
25 C   B  B   A  IF A    SA THEN 70
30 A   A - SA  PRINT C     B     A
35 IF A   1 THEN 75
40 C   B  B   A  F   INT (A/X)- X + 1
45 IF A<   X THEN A   1  GO TO 15
50 IF F>A THEN F   F  X
55 D   A  F
60 IF D   0 THEN D   INT (RND - (MA   1))
   + 1  GO TO 80
65 A   A  D  GOTO 15
70 BEEP  PRINT  PERDISTE   END
75 BEEP 1  PRINT  GANASTE   END
80 BEEP1  BEEP  PRINT  EL MAXIMO ES
 ,  MA  GOTO 15
```

Indicaciones

Al correr el programa aparecerá

MAXIMO A SACAR? ingresar el máximo de palitos a sacar por jugada

aparecera

0 0 # ? el # simboliza el numero de palitos para el juego es el turno del jugador y deberá ingresar cuantos palitos desea sacar

aparecera

\# / # # es el turno de la calculadora y los # representan los palitos segun

- palitos dejados por el jugador en su ultima jugada
- palitos dejados por la calculadora en su jugada pasada
- palitos dejados por el jugador en su jugada pesada

al cabo de unos 5 segundos aparecerá el nuevo mensaje correspondiéndole al jugador su turno

\# # # ? el jugador deba ingresar los palitos que desea sacar

- palitos dejados por la calculadora en su ultima jugada
- palitos dejados por el jugador en su jugada pasada
- palitos dejados por la calculadora en su jugada pasada

El corazón de este programa es la idea de dejar al contrario en un numero tal que sea un numero de la serie $X_n = 1 +$ (maximo + 1)

SHARP COLGADO

Sr Director

Junto con felicitarlo por su excelente revista, debo agradecerle que hayan publicado el programa que les envie en el numero 14 ¡Muchas gracias!

Deseaba pedirles que publicaran una lista más o menos completa de PEEK S y POKE S para ATARI

Además les envio este entretenido programa de COLGADO para una computadora de bolsillo PC 1245 (Sharp)

Funciona de la siguiente manera

– Pnmeramente se ejecuta con "RUN

– Luego aparece en la pantalla tantos signos – como letras tenga la palabra.

– Al apretar ' ENTER aparece un signo ¿ Al aparecer este signo uno ingresa la letra que piensa que está en la palabra y apreta ENTER

– Etc etc etc

– Cuando uno sabe cuál es la palabra, al aparecer el signo "? escribe la palabra y presiona ENTER

Espero que a los lectores de MICROBYTE les guste

Atentamente se despide

Francisco Oelker B
Cas 3047
Concepcion

P D Ojalá sea posible publicar este programa De antemano gracias

```
10:CLEAR : PAUSE " [illegible]
   L G A D O": BEEP 5:
   RANDOM : K= RND 25
20:DIM C$(1),G$(1),D$
   (1)*14 ,I$(1)
30 FOR X=1 TO K: READ C
   $(1): NEXT X: K=0
40:B= LEN C$(1)
50:D$(1)="": FOR X=1 TO
   B
60:IF I$(X)="" LET I$(X
   )= CHR$ 95
70:IF C$(1)= MID$ (C$ (1)
   ,X,1) LET I$(X)=D$(
   1
80:D$(1)=D$(1)+I$(X)
90:NEXT X
100:IF D$(1)=C$(1) THEN
    500
110:BEEP 1: WAIT 0: PRIN
    T D$(1): INPUT G
    1): K=K+1: BEEP 1
120:M= LEN G$(1)
130:IF G$(1)=C$(1) THEN
    500
140:IF M>1 THEN 300
150:GOTO 50
300:BEEP 2: PAUSE " NO
    ES ESA ": BEEP 1:
    GOTO 110
500:BEEP 3: PAUSE " ACE
    RTASTE !!": BEEP 1:
    PAUSE " EN "; STR$ K
    ;" INTENTOS": BEEP 1
510:FOR A=10 TO 30+B: H=A
    +1: G$(1)= MID$ ("LA
    PALABRA ERA "+C$(1),
    A,H): WAIT 25: PRINT
    G$ 1): NEXT A: BEEP
    1
520:PAUSE " GAME OVER
    ": BEEP 1
530:END
700:DATA "AUTOMOVILES",
    "CUERELCH","FELICITAC
    IONES","MAGNIFICOS",
    "FANTASTICOS"
710:DATA "INTELECTUALIDA
    D","PSICOLOGIA","CONS
    TITUCIONES","ACUPUNTURA
    ","CASTELLANO"
```

```
E J E M P L O:
---------

---------

---------

---------

----[illegible]

MI___OBY__

MI___OBYT_

MICRO_BYT_

MICROBYTE
ACERTASTE
```

INTERESANTES SUGERENCIAS

Sr Director

Estimados señores a quien se interesa por la computacion e intuye un explosivo auge de sus aplicaciones. en nuestro medio y que no dispone de la posibilidad de estudios profundos y sistemáticos, ni siquiera un pequeño computador para practicar la revista que ustedes editan, es el unico medio que lo relaciona con esta apasionante actividad

En calidad de lector y aficionado me permito formular las siguientes sugerencias

Detectar en forma válida las necesidades del porcentaje mas alto de lectores, para onentar el contenido de la publicación Encuentro exagerado el espacio destinado a NOTICIAS

Sería muy atractivo sortear entre los lectores un microcomputador quizas auspiciado por alguna firma comercial

Felicito a cada una de las personas que han creado y producen la revista MICROBYTE, que cumple una importante función de difusion de la computacion herramienta de la ciencia la educacion, del uso del tiempo libre en algo comparable al ajedrez y mucho mas

C Charnorro P

Consideramos con mucho interés sus sugerencias Estamos desarrollando un cuestionario para detectar los tópicos de mayor interés entre los lectores el que esperamos tenga respuesta del mayor numero de personas